



Université
de Toulouse

THÈSE

**En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :
Systèmes industriels

Présentée et soutenue par :
Karim ISHAK

le : jeudi 4 novembre 2010

Titre :

Architecture distribuée interopérable pour la gestion des projets multi-sites.
Application à la planification des activités de production

JURY

M. Patrick Charpentier, Professeur à l'ENSTIB, Université Nancy 1, Président du Jury
M. Philippe CHARBONNAUD, Professeur à l'ENI de Tarbes, Directeur de thèse
M. Bernard ARCHIMEDE, Professeur à l'ENI de Tarbes, Co-directeur de thèse
M. David CHEN, Professeur à l'Université de Bordeaux 1, Rapporteur
M. Samir LAMOURI, Professeur à l'ENSAM ParisTech, Rapporteur

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
Laboratoire Génie de Production, Ecole Nationale d'Ingénieurs de Tarbes, France

Directeur(s) de Thèse :
M. Philippe CHARBONNAUD, Professeur à l'ENI de Tarbes (directeur de thèse)
M. Bernard ARCHIMEDE, Professeur à l'ENI de Tarbes (co-irecteur de thèse)

Rapporteurs :
M. David CHEN, Professeur à l'Université de Bordeaux 1
M. Samir LAMOURI, Professeur à l'ENSAM ParisTech

« Une thèse est un travail aussi grand que vous ne pouvez pas la rendre parfaite dans une durée finie ».

Joe Wolfe

Ecole de Physique

Université de Nouvelle-Galles du Sud

Sydney Australie

A mes parents « Em Karim » et « Abou Karim »

A mes frères « Wassim » et « Jony »

A toute ma famille ...

Remerciements

Mes remerciements s'adressent en premier lieu à *Monsieur Philippe CHARBONNAUD*, directeur de thèse et *Monsieur Bernard ARCHIMEDE*, co-directeur de thèse. Leurs grandes compétences et précieux conseils ont permis l'accomplissement de ce travail. Leur confiance en mes capacités a contribué à me faire aimer ces quatre ans de recherche. Je les remercie chaleureusement pour leur compréhension et leur contact constructif, si généreux sur le plan humain comme scientifique.

Je remercie chaleureusement *Monsieur Samir LAMOURI* et *Monsieur David CHEN* pour avoir accepté de rapporter cette thèse. Je leur exprime toute ma reconnaissance pour l'intérêt porté à ce travail. Leurs remarques ont apporté matière à ma réflexion.

Je suis, également, très sensible à l'honneur que m'a fait *Monsieur Patrick CHARPENTIER*, en acceptant de présider mon jury.

Cette thèse a été réalisée au Laboratoire Génie de Production (LGP) de l'ENI de Tarbes dirigé par *Monsieur Daniel NOYES* que je remercie également.

Je remercie tous les membres du LGP, permanents et doctorants pour les moments partagés ensemble. Je remercie notamment mes compatriotes *Nassim, Rafic, et Toufic*, ainsi que mon ami *Mickael* pour leur sympathie et leur encouragement.

Je remercie ma copine *Julie* d'avoir été à côté de moi pendant ces quatre années en partageant tous les moments.

Je remercie ma mère qui a traversé la méditerranée en venant du Liban pour assister à ma soutenance de thèse et partager avec moi la joie de ce moment.

Je ne peux pas oublier mes compatriotes, les membres de l'association LIBSO, qui m'ont honoré par leur présence le jour de ma soutenance. Merci beaucoup à *Viviane, Joseph, Pierre, Michel, Jean, Mohamad, ...*

Je n'oublierai jamais de remercier la personne qui m'a accueilli et m'a hébergé pendant mes six premiers mois en France, merci beaucoup *Chantal ARBELOA* pour tout...

Je remercie tous mes amis de Toulouse et ailleurs qui de proche ou de loin m'ont soutenu pendant ces années.

Enfin, je remercie infiniment toute ma famille qui m'a tout donné pour arriver jusqu'à ce niveau là.

Table des matières

Introduction générale	1
1 Problématique et objectifs	1
2 Contributions de la thèse	3
3 Organisation du document	4

Chapitre I Problématiques des entreprises dans les marchés de production multi-site 7

1 Introduction	7
2 Réseaux d'entreprises comme support de la production multi-site	8
3 Approches traditionnelles pour la production multi-site	10
4 Approches basées sur la technologie d'agents	12
4.1 Modèle Contract Net pour la négociation entre les agents	13
4.2 SCEP : un modèle multi-agent générique pour la gestion des projets multi-sites	15
5 Discussion sur les approches de production	17
6 Places de marchés électroniques pour la production multi-site	18
6.1 Marchés orientés-projets	18
6.2 Difficultés des entreprises dans l'intégration de ces marchés	20
6.3 Marché orienté-service	21
6.4 Caractéristiques du marché orienté-service par rapport au marché orienté-projet	23
6.5 Difficulté de la gestion des projets multi-sites dans un marché orienté-service	24
7 Présentation générale et propriétés du modèle SOA	27
8 Conclusion	30

Chapitre II Architecture interopérable SCEP-SOA pour la gestion de projets multi-sites 31

1 Introduction	31
2 Définitions des concepts clés dans SCEP-SOA	31
3 Composantes de l'architecture SCEP-SOA	34
3.1 Structure du registre de services	34
3.2 Structure du producteur	35
3.3 Structure du client	37
3.4 Interactions entre les acteurs dans SCEP-SOA	39
4 Fonctionnement de l'architecture SCEP-SOA	40
4.1 Description et publication des services	41
4.2 Découverte des services et description des projets clients	42
4.3 Initialisation du système de gestion de projet chez le client	43
4.4 Interaction et gestion des projets	45

4.4.1	Choix de la granularité du service	45
4.4.2	Stratégie d'invocation des services	46
4.4.3	Collecte et transmission des contraintes de projets aux producteurs	48
4.4.4	Traitement des projets par les producteurs et envoi des résultats au client	50
4.4.5	Traitement et validation des résultats par le client	53
5	Conclusion	55

Chapitre III Interopérabilité sémantique dans SCEP-SOA 57

1	Introduction	57
2	La sémantique dans les problèmes d'interopérabilité	58
3	Ontologie et représentation de la sémantique	59
3.1	Définitions et langages pour l'ontologie	59
3.2	Classification des ontologies	61
3.3	Apports des ontologies dans la gestion interopérable de la connaissance	64
4	Problématique de l'interopérabilité sémantique dans SCEP-SOA	65
4.1	Classification de l'hétérogénéité entre les ontologies	66
4.2	Hétérogénéités des ontologies dans SCEP-SOA	69
5	Techniques d'interopérabilité des ontologies	70
5.1	Correspondance d'ontologies	71
5.2	Alignement d'ontologies	72
5.3	Fusion d'ontologies	73
5.4	Traduction d'ontologies	74
6	Approches d'intégration des ontologies	75
7	Stratégies d'interopérabilité sémantique dans SCEP-SOA	77
7.1	Stratégie basée sur l'approche multi-ontologie	77
7.2	Stratégie basée sur l'approche hybride	83
8	Comparaison des stratégies étudiées	89
9	Conclusion	93

Chapitre IV Application de SCEP-SOA à l'interopérabilité des systèmes de planification hétérogènes 95

1	Introduction	95
2	Description générale du cas d'étude traité concernant deux producteurs et un client ayant deux projets	96
3	Outils de planification utilisés par les partenaires	98
3.1	Présentation de l'ontologie de R@mses	99
3.2	Présentation de l'ontologie de TAPAS	100
3.3	Hétérogénéités entre R@msesOntology et TAPASontology	101
4	L'ontologie globale OntoBase	103
5	Modèle de l'invocation du service basée sur OntoBase	106
6	Interactions entre les partenaires et planification des projets	109

6.1	Récupération des données des projets par les agents ambassadeurs _____	111
6.2	Traduction et envoie des données aux producteurs _____	113
6.3	Réception, traduction et enregistrement du projet chez les producteurs _____	115
6.4	Planification et stockage du résultat _____	119
6.5	Traduction et renvoie du résultat au client _____	120
6.6	Traduction et dépôt du résultat dans l'environnement SCEP du client _____	122
6.7	Etude et validation du résultat proposé _____	123
7	Conclusion _____	126
Conclusion générale et perspectives _____		127
1	Conclusion générale _____	127
2	Perspectives _____	128
Références bibliographiques _____		131

Liste des figures

Chapitre I

Figure I.1. Etapes du protocole Contract Net (d'après Touaf, 2005)	14
Figure I.2. Le modèle SCEP	15
Figure I.3. Modèles SCEP organisés en réseau	16
Figure I.4. Modèle conceptuel d'un marché orienté-projet	19
Figure I.5. Modèle du marché orienté-service	22
Figure I.6. Participation simultanée d'un producteur à plusieurs EVs	25
Figure I.7. Principaux acteurs dans SOA	27
Figure I.8. Concepts clés de SOA et leurs relations (adaptée de (Nickull, 2005))	28

Chapitre II

Figure II.1. Concepts clés de SCEP-SOA et leurs relations	32
Figure II.2. Registre de services dans SCEP-SOA	34
Figure II.3. Producteur dans SCEP-SOA	35
Figure II.4. Client intégrant le modèle SCEP	38
Figure II.5. Interactions entre les acteurs dans SCEP-SOA	39
Figure II.6. Diagramme de cas d'utilisation de SCEP-SOA	40
Figure II.7. Description et publication des services	41
Figure II.8. Découverte des services et description des projets	42
Figure II.9. Système de gestion de projets basé sur SCEP	43
Figure II.10. Initialisation des activités des projets clients dans l'environnement SCEP	44
Figure II.11. Problème de l'invocation simultanée des services chez deux producteurs	48
Figure II.12. Collecte et transmission des contraintes au producteur	48
Figure II.13. Récupération des contraintes de l'environnement SCEP par les agents ambassadeurs	49
Figure II.14. Invocation du service et envoi du résultat	50
Figure II.15. Diagramme d'états-transitions des projets et de leurs résultats de gestion chez un producteur dans un cycle de gestion	51
Figure II.16. Traitement et validation d'un résultat chez le client	54

Chapitre III

Figure III.1. Classification des ontologies selon le niveau de granularité	62
----------------------------------------------------------------------------	----

Figure III.2. Les ontologies dans un plan, adapté de (Zhou et Kuntz, 2004)	63
Figure III.3. Proposition d'une classification pour l'hétérogénéité des ontologies	68
Figure III.4. Illustration de l'hétérogénéité structurelle dans SCEP-SOA	70
Figure III.5. Les trois approches pour l'intégration des ontologies (adapté de (Wache et al., 2001))	76
Figure III.6. SCEP-SOA intégrant les composants sémantiques ajoutés pour établir les correspondances ontologiques selon l'approche multi-ontologie	78
Figure III.7. SCEP-SOA intégrant les composants sémantiques pour raisonner selon l'approche multi-ontologie	79
Figure III.8. Processus de gestion d'un projet client selon l'approche multi-ontologie	80
Figure III.9. Diagramme de séquences pour la collecte et la transmission des contraintes selon l'approche multi-ontologie	80
Figure III.10. Composants d'un agent ambassadeur et leurs interactions	81
Figure III.11. Diagramme de séquences pour le traitement et la validation du résultat	82
Figure III.12. SCEP-SOA intégrant tous les composants sémantiques ajoutés dans cette stratégie	83
Figure III.13. SCEP-SOA intégrant les composants sémantiques ajoutés pour établir les correspondances ontologiques selon l'approche hybride	84
Figure III.14. SCEP-SOA intégrant les composants sémantiques ajoutés pour raisonner avec les correspondances ontologiques établies	85
Figure III.15. Structure du service et de la BDP chez le producteur	86
Figure III.16. Processus de gestion d'un projet multi-site	87
Figure III.17. Diagramme de séquences pour le traitement du projet et l'envoi du résultat	88
Figure III.18. SCEP-SOA intégrant tous les composants sémantiques selon cette stratégie	89

Chapitre IV

Figure IV.1. Cadre conceptuel de l'ontologie de R@mses (R@msesOntology)	99
Figure IV.2. Cadre conceptuel de l'ontologie de TAPAS (TAPASontology)	100
Figure IV.3. Hétérogénéités structurelle et terminologique entre les ontologies de R@mses et de TAPAS	102
Figure IV.4. Modèle conceptuel de l'ontologie OZONE et ses sous-domaines (d'après (Smith et Becker, 1997))	103
Figure IV.5. Classes ajoutées dans OntoBase et leurs relations	105
Figure IV.6. Modèle d'invocation du service Web chez les producteurs	106
Figure IV.7. Modèle du résultat de l'invocation du service Web chez les producteurs	108
Figure IV.8. Acteurs SCEP-SOA de l'exemple étudié et leurs composants	109
Figure IV.9. Planification souhaitée pour le projet 1 chez le client	111

Figure IV.10. Partie du fichier XML concernant la description du projet 1 décrit selon l'ontologie de R@mses	112
Figure IV.11. Partie du fichier XML concernant la description de l'OF1 du projet 1 décrite selon l'ontologie OntoBase	114
Figure IV.12. Description des ressources selon l'ontologie de TAPAS	117
Figure IV.13. Description selon l'ontologie de TAPAS des gammes des OF 1, 2 et 3 du projet 2 chez le producteur 2	118
Figure IV.14. Exemple d'un résultat d'ordonnancement chez le producteur 2	120
Figure IV.15. Résultat de planification décrit selon les deux ontologies TAPASOntology et OntoBase chez le producteur 2	121
Figure IV.16. Fichier XML du résultat de planification reçu du producteur 2 et traduit en R@msesOntology	123
Figure IV.17. Résultat de planification du projet 1 reçu du producteur 1	124
Figure IV.18. Résultat de planification du projet 1 reçu du producteur 2	124

Liste des tableaux

Chapitre III

Table III.1. Comparaison des deux stratégies dans SCEP-SOA _____	90
------------------------------------------------------------------	----

Chapitre IV

Tableau IV.1. Machines et activités chez les producteurs _____	96
Tableau IV.2. Description des ordres de fabrication de la chaise (projet 1) _____	98
Tableau IV.3. Description des ordres de fabrication de la table (projet 2) _____	98

Introduction générale

1 Problématique et objectifs

Aujourd'hui, la globalisation des échanges et la mondialisation du marché sont représentatives des orientations et du fonctionnement de l'économie internationale. Les fusions, acquisitions ou absorptions menées par les groupes afin de renforcer leurs compétences dans certains secteurs d'activité sont courantes. La tendance des entreprises à se recentrer sur certaines activités ainsi que la recherche de flexibilité et de réactivité entraînent une coopération avec les fournisseurs et les clients qui se concrétise par une logique de partage des réalisations, qu'elles soient technologiques ou logistiques (Enjalbert, 2006).

Dans ce contexte, les entreprises doivent faire face à une concurrence de plus en plus féroce avec un environnement en perpétuelle évolution. Les exigences difficiles à satisfaire des clients représentent une contrainte supplémentaire pour les industriels. Les grands groupes se sont adaptés en travaillant avec des outils similaires et compatibles entre eux, capables d'organiser des productions sur différents sites. Les petites et moyennes entreprises et industries ne peuvent pas en revanche mettre en œuvre les mêmes moyens. En effet, il est difficile pour de petites entités de travailler de la même manière, avec les mêmes moyens ou les mêmes savoirs. Les écarts culturels, géographiques, sociologiques sont autant de différences qui rendent ces coopérations délicates.

L'avènement d'Internet et des nouvelles technologies de l'information et de la communication a profité aux entreprises qui s'en sont servies pour créer des places de marchés électroniques améliorant ainsi leur compétitivité et leurs relations commerciales avec les partenaires (Eurochambres 2002 ; Park et Yang, 2006). Actuellement, ces marchés de production sont dominés par les grandes entreprises, véritables donneurs d'ordres, proposant leurs propres marchés pour la réalisation de leurs produits en exigeant souvent des contraintes qui pénalisent une grande partie des petites et moyennes entreprises (PMEs) ayant des capacités et des moyens limités (Gulledge, 2001 ; Bal et Swift, 2002 ; OECD, 2007).

Afin d'améliorer la présence de ces PME dans les marchés de production, un nouveau type de marché est apparu sous le nom de « marché orienté-service » (Li *et al.*, 2006 ; Griffin et Pesch, 2007). Il réduit les exigences fortes des donneurs d'ordres en offrant un marché plus dynamique. Ce marché met les PME au premier plan en décrivant leurs compétences et savoir-faire en terme de service et les fédérant dans un environnement partagé permettant aux grandes entreprises de solliciter les fournisseurs de services répondant à leurs besoins. Dans ce marché orienté-service, les collaborations et les coopérations entre les entreprises sont la plupart du temps à court terme où les entreprises clientes (qui veulent réaliser certains projets) sont amenées à collaborer avec plusieurs fournisseurs de services, normalement des PME. La réduction des exigences techniques et économiques des entreprises clientes et l'aspect dynamique des collaborations mettent en évidence la distribution de la gestion des projets en faisant coopérer les différents systèmes et applications de production des partenaires distants.

Les Systèmes Multi-Agents (Ferber, 1999) permettent de gérer la distribution de la décision. Néanmoins, cette distribution met en évidence la problématique d'hétérogénéité entre les différents systèmes et applications utilisés par les partenaires dans la gestion des projets. Différents modèles, différentes normes et technologies, différentes plates-formes et logiciels peuvent être mis en jeu au sein d'un même groupe de partenaires. Cela montre le besoin en interopérabilité afin de gérer ces hétérogénéités. L'interopérabilité peut être définie comme la capacité de communiquer avec des systèmes distants pour accéder et faire appel à leurs fonctionnalités (Vernadat, 1996). Bien que l'interopérabilité technologique soit assurée par le modèle SOA (Service Oriented Architecture) (Nickull, 2005), rien ne garantit que les informations échangées entre les systèmes des partenaires soient compréhensibles et bien interprétées. Cette problématique est bien connue sous le nom d'hétérogénéité sémantique et met en évidence le besoin en interopérabilité sémantique afin de garantir la bonne compréhension entre les partenaires en collaboration.

La problématique abordée dans ce travail peut être résumée par les objectifs suivants :

- Concevoir une architecture permettant la gestion distribuée des projets de production multi-sites. Cette architecture doit intégrer des mécanismes d'identification dynamique des partenaires et permettre une gestion des projets en temps réel en faisant coopérer et collaborer les différents systèmes et applications de production des partenaires.

- Assurer une collaboration et une coopération interopérables entre les partenaires indépendamment de toutes spécificités et particularités de certains partenaires, *i.e.* technologies, normes et standards, plates formes et logiciels, etc.
- Gérer l'hétérogénéité sémantique illustrée par les différences entre les concepts et vocabulaires manipulés par les partenaires en garantissant une bonne compréhension et une bonne interprétation des informations échangées.

2 Contributions de la thèse

Dans cette thèse, la contribution principale porte sur la proposition d'une architecture distribuée orientée-service pour gérer les projets de production multi-sites, assurant l'interopérabilité aux niveaux technologiques et sémantique entre les partenaires.

Afin de gérer la distribution de la gestion des projets de production multi-sites, nous nous sommes basés sur le modèle multi-agent SCEP (Superviseur, Client, Environnement, Producteur) développé au Laboratoire Génie de Production de l'ENI de Tarbes, permettant une coopération entre des agents responsables de la gestion des projets client et d'autres agents représentant les sites de production distants. Cependant, ce modèle présente certaines limites comme le manque d'un mécanisme de mise en relation des partenaires, ainsi que le manque d'interopérabilité car il ne permet pas une communication et une collaboration indépendantes des systèmes et modèles utilisés par les partenaires distants (Ishak *et al.*, 2008a ; Ishak *et al.*, 2008b).

Pour assurer cette interopérabilité, le modèle SCEP est couplé avec les concepts du modèle de référence pour l'interopérabilité SOA dans une architecture distribuée et interopérable SCEP-SOA de gestion de projets multi-sites (Ishak *et al.*, 2008c ; Ishak *et al.*, 2008d). Grâce à cette intégration, la mise en relation des partenaires est assurée par le registre SOA des services permettant ainsi à un client de découvrir et d'identifier les fournisseurs des services permettant la réalisation de ses projets. L'interaction entre le client et ses partenaires se fait par le biais des services SOA, faiblement couplés et indépendants de toutes technologies et plates-formes.

Finalement, afin de garantir la compréhension des informations échangées dans SCEP-SOA, cette dernière est enrichie par des composants sémantiques basés sur les ontologies (Gruber, 1993) capturant la sémantique et la structure des concepts manipulés par les différents

systèmes des partenaires. Les deux approches discutées dans (Ishak *et al.*, 2009a) peuvent être appliquées dans l'intégration des ontologies dans SCEP-SOA. La méthodologie adoptée est basée sur une approche consistant à utiliser une ontologie globale et commune dans l'échange des informations (Ishak *et al.*, 2009b). Chaque partenaire garde sa propre ontologie locale et établit des correspondances entre cette ontologie locale et l'ontologie globale (Ishak *et al.*, 2009d).

Cette architecture a été illustrée par un cas d'étude dont l'objectif est de montrer la faisabilité de la stratégie adoptée dans SCEP-SOA pour assurer l'interopérabilité sémantique entre des applications d'entreprises dédiées à la planification et à l'ordonnancement (Ishak *et al.*, 2009c ; Ishak *et al.*, 2010). Une ontologie globale a été proposée pour cet objectif capturant les concepts de base dans ce domaine. Les services de planification implémentés chez les partenaires sont décrits selon ces concepts de base ainsi que les informations de planification échangées entre les différents systèmes de planification.

3 Organisation du document

Ce mémoire de thèse est organisé comme suit :

- **Le chapitre 1** présente un état de l'art des approches existantes pour la gestion de la production multi-site ainsi que les marchés associées. Il étudie également les difficultés rencontrées par les entreprises, notamment les PME (Petites et Moyennes Entreprises), dans l'intégration de ces marchés. Afin de gérer cette production multi-site, les entreprises se basent le plus souvent sur des approches traditionnelles qui exploitent des modèles de production plus ou moins centralisés. Des approches alternatives basées sur les concepts d'agents ont été proposées pour prendre en compte le contexte distribué de la production. Une étude de ces différentes approches est présentée. Les marchés orientés-projets ainsi que les difficultés rencontrées par les PME pour l'intégration de ces marchés sont ensuite discutés. Le marché orienté-service offre un marché plus équitable et plus dynamique. Les propriétés de ce marché et la difficulté de gestion des projets dans un tel marché sont aussi discutés et suivis d'une illustration de ce marché par le modèle SOA.
- **Le chapitre 2** décrit l'architecture SCEP-SOA que nous proposons pour la gestion de projets multi-sites. Elle intègre les concepts du modèle multi-agent générique SCEP et

ceux du modèle SOA. La composition des différents acteurs (client, producteur, registre) ainsi que le fonctionnement de l'architecture SCEP-SOA sont discutés en focalisant notamment sur les interactions mises en jeu entre un client et ses partenaires lors de la gestion de ses projets. La principale limite de l'architecture telle qu'elle a été proposée concerne la problématique de l'hétérogénéité sémantique. En effet, SCEP-SOA ne garantit pas la bonne compréhension des informations échangées entre les partenaires.

- **Le chapitre 3** a pour objectif l'intégration de l'interopérabilité sémantique dans l'architecture SCEP-SOA. Afin de représenter la sémantique d'une manière interprétable, les ontologies se présentent comme des outils permettant de capturer la connaissance et les concepts d'un domaine d'une manière formelle. Une étude détaillée portant sur la définition, la classification et l'intérêt des ontologies est présentée dans ce chapitre. Du fait que chaque partenaire peut avoir sa propre ontologie, les hétérogénéités qui peuvent exister entre les ontologies dans SCEP-SOA sont aussi discutées. Deux stratégies sont proposées pour assurer l'interopérabilité sémantique dans SCEP-SOA. Elles sont basées sur les approches existantes pour l'intégration des ontologies dans les architectures distribuées. Elles sont ensuite comparées pour privilégier la plus adaptée à l'architecture SCEP-SOA.
- Enfin, **le chapitre 4** illustre le fonctionnement du modèle SCEP-SOA et de la stratégie sémantique adoptée sur un cas d'étude dédiée à la planification des projets multi-sites dans un marché de fabrication orienté-services. Ce cas d'étude s'articule autour de deux outils d'ordonnancement différents, à savoir R@mises (Re@ctive Multi-site SystEm for Scheduling) (Archimède *et al.*, 2003) et TAPAS (The Almost Perfect Approach to Scheduling) (Moutarlier *et al.*, 2000). R@mises est basé sur le modèle SCEP et est implémenté du côté du client désirant planifier ses projets. TAPAS est mis en place chez les partenaires de ce client encapsulant la fonction de l'ordonnancement dans un service SOA interopérable. L'étude présentée montre les hétérogénéités sémantiques existantes entre les ontologies de planification manipulées par SCEP et TAPAS, et propose une ontologie globale de planification utilisée dans les interactions et l'échange d'informations entre ces différents systèmes. L'étude des interactions entre les partenaires dans les différentes étapes du processus de planification des projets clients illustre l'intérêt de la stratégie sémantique proposée dans SCEP-SOA.

Chapitre I

Problématiques des entreprises dans les marchés de production multi-site

1 Introduction

De manière à demeurer compétitives, les entreprises externalisent les activités qui ne sont pas au cœur de leur métier, ou celles pour lesquelles elles ne possèdent pas d'avantages concurrentiels. Ceci les conduit en particulier à délocaliser la plupart de leurs activités à faible valeur ajoutée. Cette externalisation des activités se manifeste sous la forme de production multi-site, permettant à l'entreprise de se concentrer sur son propre savoir-faire et de partager les autres activités avec d'autres sites ou entreprises. Par conséquent, plusieurs réseaux d'entreprises regroupant les entreprises partenaires sont créés avec des durées de collaboration plus ou moins courtes. Ce regroupement d'entreprises en réseau nécessite une collaboration et une coopération entre les différents partenaires pour gérer la réalisation des projets multi-sites.

Afin de gérer cette production multi-site, les entreprises se basent le plus souvent sur des approches traditionnelles qui exploitent des modèles de production plus ou moins centralisés. Des approches alternatives basées sur les concepts d'agents ont été proposées pour prendre en compte le contexte distribué de la production. L'avènement d'Internet et des nouvelles technologies de l'information et de la communication permet aujourd'hui aux entreprises de créer des places de marchés électroniques pour améliorer la gestion de la production multi-site et ainsi leur compétitivité.

Bien que les techniques pour appréhender la production multi-site évoluent, il n'en demeure pas moins qu'il existe encore de nombreux points d'amélioration. L'objectif de ce chapitre est de faire un Etat de l'Art des approches existantes pour la gestion de la production multi-site et de ses types de marchés en étudiant les difficultés rencontrées par les entreprises, notamment les PME (Petites et Moyennes Entreprises), dans l'intégration de ces marchés. Dans un premier temps, les différentes organisations d'entreprises en réseau sont discutées dans la

section 2. Les approches et modèles traditionnelles et celles basées sur les agents pour la gestion de la production multi-site sont ensuite présentées dans les sections 3 et 4. Une discussion sur ces approches est donnée dans la section 5. Les marchés orientés-projets et orientés-services sont discutés dans la section 6 qui présente aussi les problématiques identifiées dans les collaborations entre les partenaires de ces marchés. La section 7 décrit le modèle SOA (Service Oriented Architecture) illustrant l'implémentation d'un marché orienté-service.

2 Réseaux d'entreprises comme support de la production multi-site

La production multi-site peut être réalisée par des réseaux ou organisations d'entreprises qui diffèrent notamment par leur horizon temporel ou leur durée de vie. La globalisation du marché et les opportunités d'affaires ont poussé les entreprises à travailler ensemble dans des organisations temporaires. Plusieurs options stratégiques peuvent être adoptées et peuvent générer une variété de formes ou modèles d'organisation d'entreprises illustrant la production distribuée (LoNigro *et al.*, 2006). Parmi ces formes, nous avons les chaînes logistiques, les entreprises étendues, et les entreprises virtuelles.

Une chaîne logistique SC (Supply Chain) peut se définir par référence à un produit fini donné pour lequel l'entreprise est appréhendée du point de vue du rôle qu'elle joue dans le processus de transformation du produit. Un autre point de vue inclut l'ensemble des chaînes logistiques dans lesquelles est impliquée une entreprise, mais se limite parfois aux fournisseurs et aux clients immédiats de l'entreprise (Monsarrat *et al.*, 2005). Pires *et al.*, (2001) considèrent qu'une SC en général est un réseau de compagnies autonomes ou semi-autonomes qui sont responsables collectivement de l'obtention, la production et la délivrance d'une ou de plusieurs familles de produits. La gestion d'une chaîne logistique ou SCM (Supply Chain Management) est une gestion du flux de produits et de services et du flux d'informations associées dans tout le système d'affaire pour maximiser la valeur pour le consommateur final (Kovacs et Paganelli, 2003). L'objectif principal de la SCM est de maximiser la synergie entre toutes les parties de la chaîne de demande afin de servir effectivement et à moindre coût le consommateur final.

La prise de décision dans une SC est divisée en plusieurs catégories : décisions stratégiques (niveau plus élevé et avec une longue durée, quels produits doivent être produits et dans

quelles usines), décisions tactiques (durée de temps moyenne, planification et ordonnancement à long terme), décisions opérationnelles (durée courte, activités journalières et planification à niveau bas). Malgré les intérêts communs partagés entre tous les partenaires participant à une SC, la distribution de la décision reste une problématique du fait que les contextes et les objectifs propres aux différents acteurs de la chaîne peuvent différer (Monsarrat *et al.*, 2005). Pour résoudre cette problématique et parvenir à un compromis satisfaisant, les acteurs doivent coopérer. D'autre part, lors de l'implémentation d'une SC, les problèmes identifiés par Sauer et Appelrath, (2002) peuvent se résumer à l'absence d'un plan singulier pour l'organisation toute entière (les organisations ont leurs propres objectifs qui sont souvent conflictuels), à un flux d'informations plus important que le flux matériel, à une grande masse d'informations à échanger nécessitant un certain degré de confiance entre les compagnies sur les informations fournies : actuelles, complètes, et correctes.

Aujourd'hui, les entreprises choisissent de regrouper leurs activités au sein de pôles d'excellence, aussi n'est-il pas rare de produire une pièce sur plusieurs sites afin d'en augmenter la productivité. Compte tenu des demandes changeantes des clients, les configurations de ces réseaux varient fréquemment et les capacités de production sont soumises à des variations inhérentes aux performances des partenaires du réseau d'entreprises. Ces phénomènes font que les réseaux d'entreprises sont en perpétuelle évolution et qu'il est difficile d'organiser de telles entités.

De nouvelles définitions sont venues caractériser les différents types d'organisation, *i.e.* les entreprises étendues et les entreprises virtuelles. Des réseaux formés par une entreprise et l'ensemble de ses partenaires directs, *i.e.* les clients, les fournisseurs et les prestataires, sont considérés comme un tout intégré devant fonctionner comme une entreprise unique. Ces réseaux de co-traitants et sous-traitants forment des entreprises étendues. L'entreprise étendue (EE) est la formation d'une proche coordination dans la conception, le développement, le calcul des coûts et la coordination des ordonnancements respectifs de chaque entreprise en faisant coopérer des entreprises manufacturières indépendantes et leurs fournisseurs associés. Le concept d'entreprise étendue est en partie né des essais des entreprises, géographiquement réparties, de construire des partenariats pour gagner en compétitivité (Jagdev et Browne, 1998). Ce concept s'étend à toutes les entreprises, aux services ainsi qu'aux ressources extérieures qui collaborent. Les buts sont multiples : réduction des durées de fabrication,

diminution des temps de communication, baisse des durées de développement de nouveaux produits, création d'organisations plus efficaces, etc.

Si l'entreprise étendue est une coopération stable pour développer des relations à long terme avec sous-traitants et fournisseurs, l'entreprise virtuelle (EV) est plutôt un réseau temporaire de compagnies indépendantes pour réaliser un produit spécifique et disparaître avec lui. Bien qu'elle regroupe plusieurs entreprises, cette EV apparaît comme une compagnie singulière sur le marché (Pires *et al.*, 2001). Ses principales caractéristiques peuvent être résumées par la concentration sur les cœurs de métiers, la coopération temporaire, la structure dynamique, et la confiance. Sauer *et al.*, (2000) considèrent qu'une EV représente un réseau de compagnies engagées pour accomplir une tâche particulière. Une EV peut être vue comme une SC temporaire. Le cycle de vie d'une EV se résume par sa création, son opération d'activité et sa dissolution (Pires *et al.*, 2001). La phase de création consiste à regrouper les entreprises participantes. La coopération entre les partenaires constitue la phase d'activité. La fin de la coopération entraîne la dissolution de l'EV.

Pour résumer, les organisations présentées précédemment peuvent être réparties en deux catégories selon l'horizon de travail et la durée de vie des collaborations entre les partenaires au sein de cette organisation. La première catégorie concerne les organisations ayant une durée de vie plus ou moins longue et des collaborations plus stables telles que la SC et l'EE. La deuxième concerne des relations temporaires à court terme entre des entreprises partenaires, moins stables que celles dans la première catégorie, telle que l'EV.

Pour mieux gérer la réalisation des projets au sein de ces différentes organisations, les entreprises utilisent des systèmes et des applications de production basés sur des approches plus ou moins centralisées.

3 Approches traditionnelles pour la production multi-site

Dans les années 80, le CIM (Computer Integrated Manufacturing) a permis une première intégration des activités de production (François, 2007). La généralisation du concept d'intégration de systèmes informatiques et des processus métier dans tous les domaines de l'entreprise a donné naissance aux ERP (Enterprise Resource Planning), extension du terme MRP (Material Requirements Planning) qui vise à planifier conjointement, à partir des demandes fermes des clients ou de leurs prévisions, les besoins en composants ainsi que les

ordres de fabrication (Bourrières *et al.*, 2005). Un ERP fournit à l'ensemble des acteurs de l'entreprise une image unifiée, intégrée, cohérente et homogène de l'ensemble des informations dont ils ont besoin (Botta-Genoulaz, 2005). Cependant, les progiciels de type ERP visent à une centralisation de l'information. Ce système transactionnel demande désormais d'être complété par des systèmes décisionnels ou d'aide à la décision dans un but d'optimisation dépassant les frontières de l'entreprise.

D'autres progiciels spécialisés et dédiés à la gestion de la chaîne logistique, gravitant toujours autour des ERP, ont été développés : CRM (Customer Relationship Management) une application de Gestion de la Relation Client qui regroupe toutes les fonctions permettant d'intégrer les clients dans le système d'information de l'entreprise, SCM (Supply Chain Management) une application de Gestion de la Chaîne Logistique qui regroupe toutes les fonctions permettant d'intégrer les fournisseurs et la logistique au système d'information de l'entreprise, HRM (Human Resource Management) une application de Gestion des Ressources Humaines qui permet d'offrir toutes les fonctions permettant la gestion du personnel de l'entreprise, etc. A ceux-ci s'ajoutent encore des solutions PLM (Product Lifecycle Management) permettant la gestion et la maîtrise de la complexité qui caractérise le développement et le suivi des produits afin de réduire les coûts et les délais et à augmenter la qualité des produits (Paviot, 2010), et des logiciels PDM (Product Data Management) (Eynard *et al.*, 2004) qui offrent des fonctions permettant de gérer les données relatives à la conception et à la fabrication des produits dans un contexte PLM.

Les APS (Advanced Planning Systems) offrent des fonctionnalités alternatives à celles des ERP dans le domaine de la planification des activités de production. Ces systèmes peuvent notamment prendre en compte la capacité finie des ressources lors de la planification, simuler plusieurs scénarios de planification et gérer plusieurs sites de production, ce qui est nécessaire dans le cas des entreprises multi-sites ou des réseaux d'entreprises. En effet, l'apport des APS à la production relève de deux caractéristiques qui sont l'utilisation d'algorithmes mathématiques d'optimisation et l'élargissement de la gestion interne des entreprises à une gestion globale de la chaîne logistique (Telle, 2003). Les APS peuvent être déployés dans des environnements dynamiques du fait qu'ils offrent un mécanisme rapide pour le calcul des plans en offrant un plan optimal pour une chaîne logistique complète.

Afin de mieux prendre en compte l'aspect distribué et dynamique dans la gestion de la production multi-site et d'exploiter des modèles moins agrégés, des approches alternatives basées sur les agents ont été proposées.

4 Approches basées sur la technologie d'agents

Un système multi-agent (SMA) est un système d'intelligence artificielle composé de population d'agents autonomes qui coopèrent entre eux pour atteindre des buts communs, tout en poursuivant simultanément différents objectifs (Jinghua et Wenjian, 2005). Dans la littérature, plusieurs modèles basés sur les SMAs ont été proposés dans plusieurs domaines de la production multi-site. Dans le domaine de la planification distribuée, un modèle SMA pour la planification distribuée de la production a été proposé dans (Lima *et al.*, 2006). Il est basé sur trois caractéristiques : la distribution, l'autonomie, et la reconfiguration, en faisant coopérer un ensemble d'agents via un tableau noir (blackboard) qui est un espace Java (Javaspaces). Dans le même domaine, un système de gestion d'ordonnancement intensif de production a été proposé dans (Nishioka, 2004). Il est basé sur des agents qui fonctionnent d'une manière autonome pour leurs problèmes locaux et en même temps leurs activités sont synchronisées pour achever l'objectif global de la fabrication de l'entreprise. Dans un contexte similaire, un système multi-agent pour la planification et la modélisation de la production est proposé dans (Lazansky *et al.*, 2001). Le système proposé remplace le processus de planification de la production centralisé par un processus de négociation entre les agents autonomes coopératifs.

Dans le domaine de la gestion et la coordination des chaînes logistiques et des entreprises virtuelles, un système de prise de décision pour la planification logistique d'une usine est présenté dans (Baumgärtel *et al.*, 1996), qui est basé sur les concepts des SMAs. La structure de l'usine et le processus de fabrication sont modélisés par un SMA en représentant chaque centre de fabrication par un agent. Cette approche permet d'encapsuler la structure d'un centre de fabrication et sa stratégie d'optimisation spécifique dans un agent, augmentant ainsi la modularité du système. Dans le même contexte, des algorithmes de prise de décision dans la gestion d'une chaîne logistique ont été proposés dans (Kovalchuk, 2008). Le projet concerne principalement la partie demande de la chaîne logistique. En particulier, différentes méthodes pour prévoir des prix d'offres qui peuvent exister dans les ordres des clients (gain des prix demandés) sont explorées et comparées dans le système. Une modélisation d'une EV basée sur les agents a été proposée dans (Oprea, 2005). Dans son étude, l'auteur identifie

certaines obstacles dans l'implémentation d'un SMA mettant en évidence le manque de certaines caractéristiques importantes qui sont des facteurs cruciaux dans une EV réelle, tels que la robustesse, les mécanismes de sécurité, des standards d'interopérabilité, ainsi que des interfaces faciles avec les systèmes anciens (legacy systems) des entreprises participant à l'EV. Thibaud *et al.*, (2007) développent une nouvelle approche pour gérer la planification des ressources multi-sites afin de coordonner le besoin du réseau d'entreprises participant à l'EV. Le composant principal de ce modèle est appelé nœud de l'entreprise virtuelle ou VEN (Virtual Enterprise Node) représentant un partenaire en contact avec plusieurs fournisseurs et plusieurs clients.

Dans les SMAs, la négociation est définie comme un processus par lequel une décision est atteinte par deux ou plusieurs agents, chacun essaie d'atteindre un but ou un objectif individuel (Marinova, 2002). En général, le processus de négociation dans les SMAs est divisé en trois phases consécutives : invitation, offre, et attribution (Jiao *et al.*, 2006). Le protocole Contract Net (CNP pour Contract Net Protocol) (Smith, 1980) définit les messages de base en accord avec ces phases et la manière de traiter chaque message.

4.1 Modèle Contract Net pour la négociation entre les agents

Le modèle Contract Net propose une technique d'allocation de tâches dédiée à la résolution distribuée de problèmes. Il s'agit d'un ensemble d'agents qui peuvent passer des contrats selon un protocole fixe qui a été imaginé à partir du protocole d'appel d'offres des contrats publics comme illustré sur la figure I.1. L'agent ayant des tâches à accomplir (maître d'ouvrage), fait un appel d'offres (annonce de tâches). L'annonce est diffusée à l'ensemble des agents. Les autres agents évaluent ces tâches. Ceux pouvant en réaliser une font des propositions (offres) et deviennent alors des contractants potentiels. Le manager évalue ces offres et décide de passer un contrat avec l'un de ces agents.

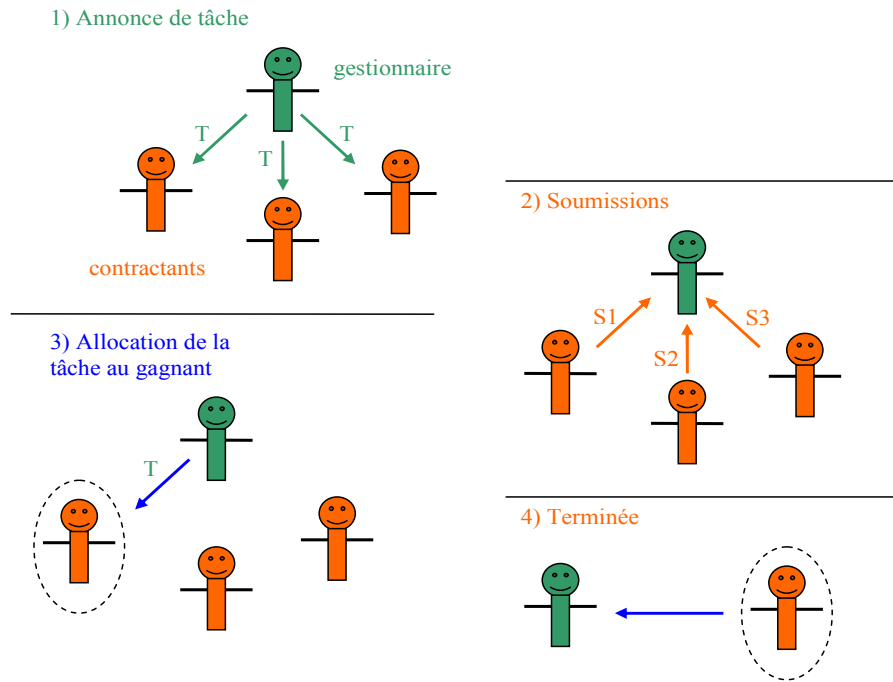


Figure I.1. Etapes du protocole Contract Net (d'après Touaf, 2005)

Dans le CNP, les agents peuvent prendre deux rôles : gestionnaire et contractant. L'agent qui doit exécuter une tâche (le maître d'ouvrage) commence par décomposer cette tâche en plusieurs sous-tâches. Le maître d'ouvrage annonce chaque sous-tâche sur un réseau d'agents (les contractants). Les agents qui reçoivent une annonce de tâches à accomplir évaluent l'annonce. Les agents qui ont les ressources appropriées pour accomplir la tâche envoient au maître d'ouvrage des offres indiquant leurs capacités à réaliser la tâche. Le maître d'ouvrage rassemble toutes les propositions qu'il a reçues et alloue la tâche à l'agent qui a fait la meilleure proposition. Ensuite, le maître d'ouvrage et les contractants échangent les informations nécessaires durant l'accomplissement des tâches. Par exemple, le contractant annoncera au maître d'ouvrage quand l'exécution de la tâche sera terminée. Dans des cas exceptionnels, un maître d'ouvrage peut annuler le contrat : annoncer au contractant qu'il faut abandonner l'exécution de la tâche.

Les rôles des agents ne sont pas spécifiés à l'avance. N'importe quel agent peut agir comme un maître d'ouvrage et lancer un appel d'offres. N'importe quel autre agent peut être un contractant pour une annonce qu'on a faite. Cette flexibilité permet de nouvelles décompositions des tâches. Un contractant pour une tâche spécifique peut agir comme un gestionnaire en décomposant sa tâche et en annonçant les sous-tâches à d'autres agents. Les liens entre les gestionnaires et les contractants pour diverses tâches et sous-tâches forment un réseau contractuel hiérarchique qui permet la division des tâches et la synthèse des résultats.

Ce protocole Contract Net a été amélioré par le FIPA (Foundation for Intelligent Physical Agents) pour gérer plusieurs négociations en parallèle en ajoutant des messages de refus ou de confirmation pour les interactions entre agents (FIPA, 2000).

4.2 SCEP : un modèle multi-agent générique pour la gestion des projets multi-sites

Le modèle multi-agent SCEP (Superviseur, Client, Environnement, Producteur) est un modèle générique permettant de résoudre d'autres problèmes que ceux de planification et d'ordonnancement. Il propose une architecture client/serveur pour résoudre des problèmes de grande taille nécessitant la puissance de calcul de plusieurs processeurs. Ce modèle distribué a été proposé pour résoudre les problèmes d'ordonnancement dans (Archimède et Coudert, 2001a ; Archimède et Coudert, 2001b) introduisant une coopération indirecte entre des agents Clients et des agents Producteurs comme illustré sur la figure I.2.

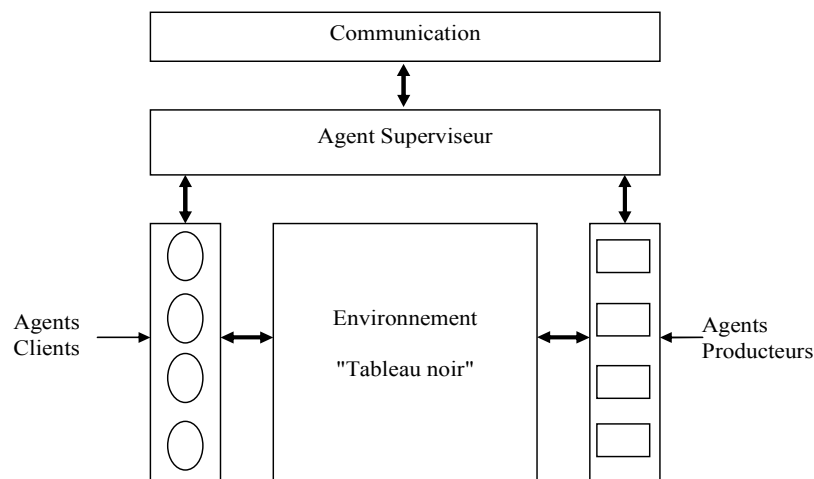


Figure I.2. Le modèle SCEP

Cette coopération est réalisée via un environnement de type "Tableau noir" sous le contrôle d'un agent Superviseur. Le "Tableau noir" est le lieu d'échange des messages durant les négociations entre les agents. L'agent superviseur lance d'abord les agents clients et demande aux agents producteurs d'attendre. Chaque agent client représente un projet à réaliser et effectue ses actions, indépendamment du comportement des autres agents. Dans le cadre de l'utilisation de SCEP pour résoudre des problèmes d'ordonnancement, les actions de l'agent client consistent en la proposition de ses contraintes souhaitées pour les activités demandées par son projet. L'agent client informe le superviseur lorsqu'il a terminé. Une fois enregistrée la dernière action du dernier agent client, le superviseur active les agents producteurs et invite les agents clients à attendre. Les agents producteurs effectuent à leur tour leurs actions et en

informe le superviseur dès qu'ils ont terminé. Pour les problèmes d'ordonnancement, ses actions consistent en des réponses aux différents agents clients qui avaient proposés leurs contraintes souhaitées. Une fois la dernière action du dernier agent producteur effectuée, l'agent superviseur redonne le contrôle aux agents clients. Cette alternance entre l'activation des agents clients et celle des agents producteurs est répétée jusqu'au positionnement final de toutes les tâches dans l'environnement.

Le modèle SCEP a montré son adaptation à la gestion distribuée des projets multi-sites. Un prototype a été réalisé illustrant son fonctionnement en réseau afin d'établir un ordonnancement distribué faisant coopérer plusieurs sites répartis géographiquement. Un exemple d'implémentation est donné sur la figure I.3.

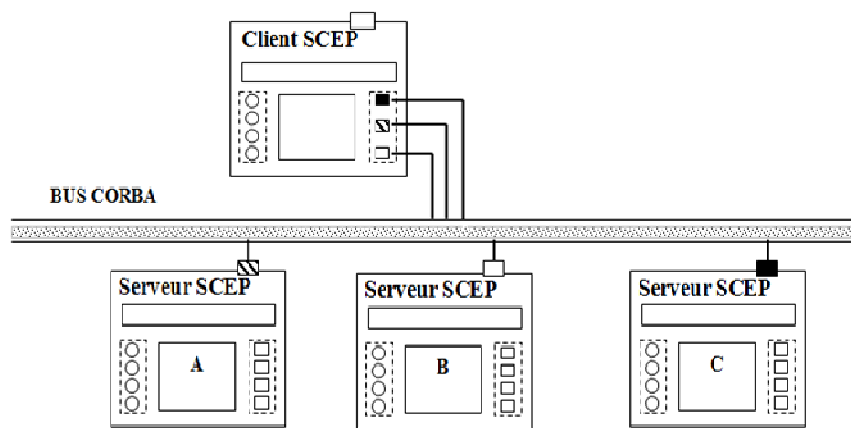


Figure I.3. Modèles SCEP organisés en réseau

Le client SCEP est une entreprise possédant quatre projets à réaliser. Ces projets sont représentés par les quatre agents clients. Trois sites (serveurs) distants sont impliqués dans la planification de ces projets. Un des sites assure le transport entre les deux autres sites considérés. CORBA est le support de transmission des informations. Le mécanisme de fonctionnement est proche du modèle Contract Net du fait qu'il considère des propositions d'offres par les sites distants qui seront étudiés et validées par la suite par le client SCEP. Néanmoins, SCEP permet une sorte de compétition entre les agents clients qui peuvent partager des ressources communes, ce qui n'est pas envisagé dans le modèle Contract Net. Les propositions des agents clients (les gestionnaires) concernant toutes les tâches sont prises en compte par les agents producteurs (les contractants) en même temps, contrairement au CNP qui les traite successivement.

5 Discussion sur les approches de production

Bien que les approches traditionnelles soient utilisées par les entreprises au sein de leurs réseaux d'entreprises, elles centralisent néanmoins la décision et la gestion des projets. Face à un tel foisonnement de progiciels et d'éditeurs, *i.e.* SAP, Peoplesoft, JD Edwards, Baan, Oracle, Siebel, Manugistics, I2 Technologies, etc., les industriels sont parfois un peu perdus quant au choix des logiciels visant à améliorer leur performance (François, 2007). De plus, l'implantation et la configuration de ces progiciels sont souvent très lourdes (en temps et en ressources), et les résultats attendus ne sont pas toujours au rendez-vous (Botta-Genoulaz et Millet, 2005). Aussi, les solutions qu'elles offrent souffrent de robustesse et restent strictement déterministes sans tenir compte de l'aspect dynamique des marchés et de la variation du processus physique (temps gamme, variabilité des fournisseurs, etc.) (Genin *et al.*, 2005).

D'autre part, les SMAs permettent de concevoir des outils et logiciels supportant la coopération tout en exploitant des modèles moins agrégés et moins abstraits. L'utilisation des SMAs dans la production multi-site offre un cadre simple de modélisation des systèmes de production ainsi qu'une distribution naturelle de la décision et un contrôle décentralisé qui garantit l'autonomie de chaque entité et la flexibilité du réseau d'entreprises impliquées dans la réalisation d'un projet de production multi-site. Les systèmes multi-agents ont montré leur adaptation à l'aspect distribué de la gestion de production multi-site du fait qu'ils permettent une distribution naturelle de la décision. Cette approche s'adapte bien à la nature des systèmes de production généralement composés de plusieurs équipements, où il faut intégrer plusieurs fonctions (conduite, gestion de production, etc.). Afin de gérer d'une manière dynamique, indépendante, et distribuée la réalisation des projets multi-sites, le modèle générique SCEP déjà présenté est bien adapté à cet objectif.

Afin de tirer avantages de ces différentes approches, de mieux gérer leurs collaborations et leurs communications, et d'améliorer leur compétitivité, les entreprises se basent actuellement sur les technologies d'informations et de communications en créant des places de marchés électroniques.

6 Places de marchés électroniques pour la production multi-site

Internet a bouleversé le commerce avec le déploiement du commerce électronique. Cette technologie de communication a poussé les grands groupes commerçants à en profiter pour améliorer leur collaboration et leur intégration dans le commerce. La notion de place de marché électronique est devenue capitale dans les marchés commerciaux. Une place de marché électronique est un site Web qui contient des fonctions de commerce pour plusieurs compagnies d'achat et de vente (Eurochambres, 2002). Elle relie les acheteurs et les vendeurs (Park et Yang, 2006). Le fournisseur de la place de marché électronique est le créateur du marché. Il n'a pas le contrôle sur les prix des produits. Les acheteurs et les vendeurs font des affaires et payent des frais pour l'utilisation du marché. Le challenge le plus important des places de marché électroniques réside dans la méthode pour attirer et retenir le plus grand nombre d'acheteurs et de vendeurs possibles. D'autres challenges visent à sécuriser une base de produits de commerce, à augmenter la connaissance des marques, à fournir un temps d'accès aux marchés et à réduire les coûts (Park et Yang, 2006).

On distingue deux types de place de marché électronique : les places de marché verticales et les places de marché horizontales. Une place de marché verticale relie les acheteurs et les fournisseurs dans un domaine précis de l'industrie. Une place de marché horizontale répond aux requêtes des acheteurs, négocie avec les fournisseurs et envoie les ordres aux vendeurs et ce pour plusieurs domaines et secteurs industriels (Popovic, 2002). Dans les places de marché électroniques horizontales, la quantité d'informations échangées entre les partenaires ainsi que le nombre de transactions effectuées sont plus importants que dans les places de marché électroniques verticales (Park et Yang, 2006). Un facteur critique pour améliorer la performance des places de marché électroniques est l'attraction des acheteurs, des vendeurs et des produits, ainsi que la flexibilité du marché et la recherche des partenaires sans oublier le critère de confiance entre les partenaires du marché.

Quel que soit le type de la place de marché horizontale ou verticale, elle peut être orientée-projet ou orientée-service.

6.1 Marchés orientés-projets

La majorité des marchés électroniques de production actuels sont dominés par les grandes entreprises reconnues sous le nom de donneurs d'ordres ou OEM (Original Equipment Manufacturer), qui créent leurs propres places de marchés orientés-projets pour la réalisation

et la gestion de leurs projets (Gulledge, 2001 ; Bal et Swift, 2002 ; OECD, 2007). A titre d'exemple Airbus a des projets différents concernant la conception des avions, la production des pièces, la maintenance, etc. Chaque projet nécessite certaines compétences et capacités de réalisation. Le marché orienté-projet est créé et contrôlé par un OEM, dans lequel celui-ci définit ses projets en termes de lots en définissant l'ensemble des contraintes exigées pour la réalisation de chaque lot. Chaque lot regroupe certaines activités faisant partie du processus de réalisation du projet considéré. Les lots sont ensuite déposés dans la place de marché comme illustré sur la figure I.4.

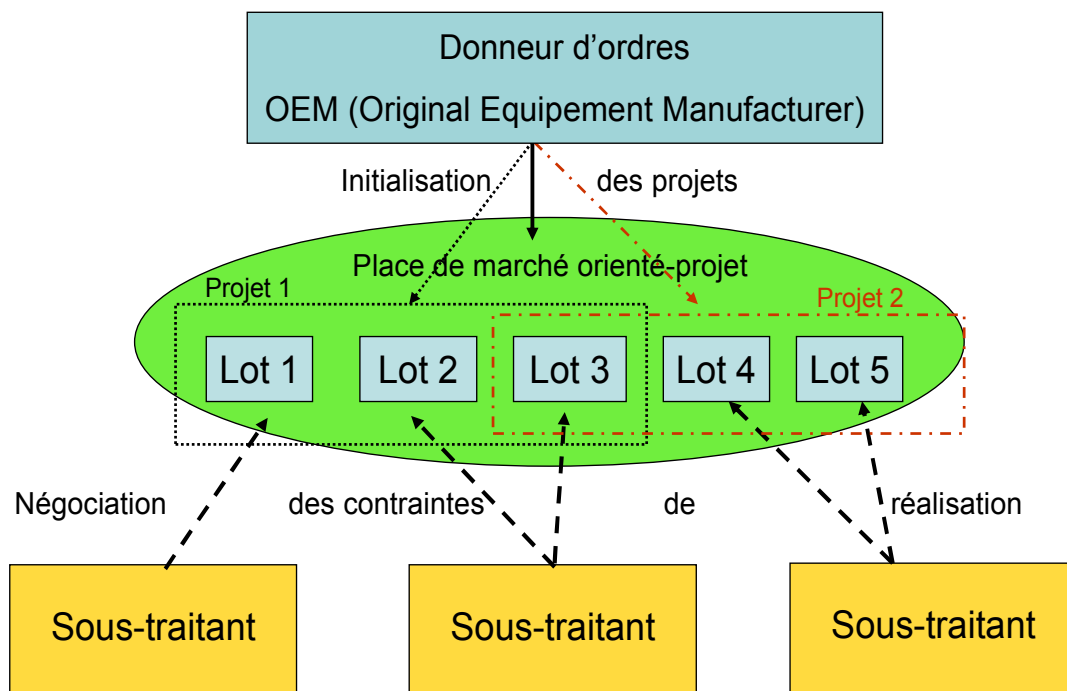


Figure I.4. Modèle conceptuel d'un marché orienté-projet

Après la définition des projets dans la place de marché, les différentes entreprises ou porteurs de projets, *i.e.* les entreprises sous-traitantes, viendront négocier avec l'OEM pour se procurer de (s) lot(s) dont ils possèdent les capacités et les compétences définies et demandées par l'OEM. En effet, pour être sélectionnée par l'OEM, un sous-traitant doit vérifier tous les critères et les contraintes demandés par l'OEM et posséder toutes les compétences et les capacités exigées par ce dernier pour les lots de projets définis. Par conséquent, un réseau d'entreprises est ainsi créé comprenant l'OEM et l'ensemble des porteurs de projets (sous-traitants) sélectionnés par cet OEM pour la réalisation des différents lots de ses projets.

Notons que plusieurs rangs de sous-traitants peuvent exister dans un tel marché. Un même sous-traitant engagé dans la réalisation d'un lot, peut lui-même être un OEM pour d'autres

sous-traitants en créant son propre marché orienté-projet pour ce lot. Dans ce marché, le sous-traitant définit les contraintes requises pour la réalisation de son projet, *i.e.* le lot de l'OEM principal, qui lui-même peut être divisé en plusieurs lots. Un OEM gère uniquement les relations avec ses partenaires ou sous-traitants de premier rang. Il ne s'occupe pas des sous-traitants des autres rangs, *i.e.* les sous-traitants des sous-traitants.

6.2 Difficultés des entreprises dans l'intégration de ces marchés

Dans un marché orienté-projet, la plupart des projets nécessitent un niveau de compétences et des capacités élevés. Actuellement, la plupart des PME ont beaucoup de difficulté à intégrer un tel type de marché. Cela est dû à leurs limites économiques et technologiques (Gulledge, 2001). Ces limites diminuent le degré de compétitivité des PME dans le marché global. Dans le contexte de la gestion d'une chaîne logistique, parmi les obstacles qu'une PME peut rencontrer, on trouve la difficulté de gérer ses fournisseurs et la grande compétition dans la chaîne ainsi que le manque de coopération. Pour appréhender ces obstacles, les auteurs dans (Udomleartprasert *et al.*, 2003) proposent une gestion à chaînes verticales (Vertical Chain Management ou VCM).

La gestion à chaînes verticales est l'outil stratégique pour la gestion de matériel et de capacité aux fournisseurs. Le principe est de combiner le volume et la compétitivité au sein de la même communauté afin de négocier pour la meilleure affaire. Le modèle a été amélioré pour réduire au minimum la faiblesse et la limitation des PME. Cependant, cette solution souffre de l'incapacité à court terme d'attirer un grand nombre d'acheteurs, de vendeurs et de produits ainsi que le problème de la confiance entre les partenaires. Ce problème a été traité dans (Ren *et al.*, 2006), où les auteurs proposent le concept d'e-Hub qui lève les barrières de la coopération entre une grande entreprise et ses fournisseurs (les PME). Le Hub est un concept spécialisé qui permet l'échange de documents entre les organisations (Gulledge, 2001). La solution proposée dans (Ren *et al.*, 2006) offre aux utilisateurs (en particulier les PME) un espace de travail unique pour la planification de projets collaboratifs et la négociation de contrats.

L'étude présentée dans (OECD, 2007) résume les différents obstacles rencontrés par les PME lors de leur intégration dans un marché orienté-projet :

- Contraintes d'intégration élevées exigées par l'OEM : c'est l'OEM, comme concepteur du marché, qui définit le niveau de capacité et de compétences pour les projets dans le marché.

Du fait que ces contraintes sont définies pour un lot de projet comprenant un ensemble d'activité demandées par le projet, cela augmente le niveau de compétence demandé pour le choix d'un sous-traitant ;

- Domination des grandes entreprises dans le marché: comme l'OEM demande des capacités et des compétences élevées pour la réalisation de ses projets, cela réduit le degré de compétitivité et permet aux entreprises satisfaisant ces contraintes fortes d'envahir le marché ;
- Capacités et compétences limitées des PME devant celles demandées par les OEMs, pénalisant les PME n'ayant pas un degré de compétitivité élevé pour entrer en compétition avec des entreprises sous-traitantes plus compétitives.

Malgré les tentatives d'amélioration et les solutions d'aides proposées dans la littérature pour les PME, l'intégration de ces dernières dans les marchés orientés-projets reste très coûteuse. Cela provient du fait qu'elles sont souvent obligées d'augmenter leur capacité et leur niveau de compétence pour pouvoir entrer en compétition et satisfaire les demandes des OEMs, impliquant un coût technologique très élevé (modification des systèmes de communication, de planification, ...). Les sous-traitants d'un OEM sont amenés souvent à modifier leurs fonctions et processus internes ainsi que leurs outils de gestion et applications utilisés afin de répondre au besoin de l'OEM et de bien respecter les contraintes exigées. Notons aussi que dans un tel environnement, les réseaux d'entreprises tendent souvent vers des SCs et des EEs plutôt que des EVs. Les modifications effectuées par les sous-traitants d'un OEM pour satisfaire les exigences de ce dernier facilitent les prochaines collaborations avec cet OEM créant ainsi un environnement de collaboration plus stable.

6.3 Marché orienté-service

Les marchés orientés-services mettent en relation plusieurs clients possédant des projets de production avec plusieurs producteurs ou fournisseurs de services permettant la réalisation des projets clients (Ghenniwa et Chen, 2004). La notion du service est essentielle dans ce type de marché. Un service regroupe un ou plusieurs savoir-faire dans un domaine de services bien défini. Un domaine de service est un espace regroupant plusieurs services permettant de réaliser un processus métier déterminé. A titre d'exemple, un domaine de service peut être le domaine de la production comprenant des services de fabrication, de transport, de conception, etc. Un service peut être élémentaire, *i.e.* n'appelle pas d'autres services, ou être composite construit à l'aide de services élémentaires (Griffin et Pesch, 2007).

Aucune relation ne lie a priori les services entre eux. Une fois les services définis dans la place de marché, les clients souhaitant réaliser leurs projets, viennent sur ce marché pour chercher les producteurs capables de réaliser leurs projets. Les clients et les producteurs ne se connaissent pas a priori. Un modèle de marché orienté-service est présenté sur la figure I.5.

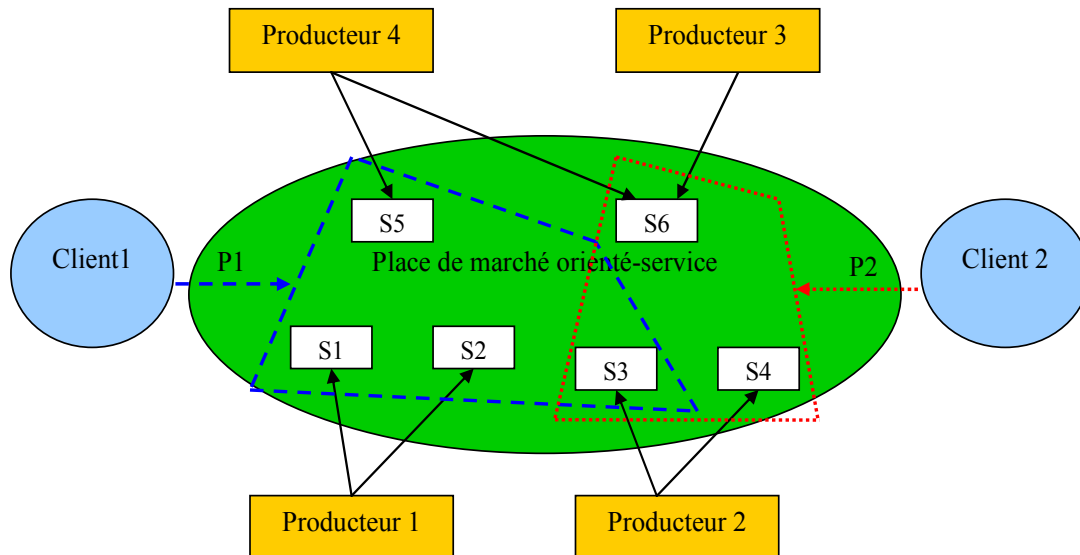


Figure I.5. Modèle du marché orienté-service

La gestion de la place de marché est réalisée par le propriétaire du marché qui doit être une entité indépendante des fournisseurs et des clients (Griffin et Pesch, 2007). Cette place doit contenir la description des services fournis par leurs producteurs et permettre aux différents clients de construire leurs projets à partir de ces services. Les fonctions de déclaration des savoir-faire et d'initialisation des projets sont essentielles pour le fonctionnement d'une place de marché orienté-service (Li *et al.*, 2006). La déclaration des savoir-faire concerne la publication des services dans la place du marché par les producteurs. Un producteur peut offrir un ou plusieurs services (cas du Producteur 1 sur la figure I.5). Un service, ou plus précisément la finalité d'un service, c'est-à-dire la fonction assurée par le service dans son domaine, peut être fourni par un ou plusieurs producteurs (cas du service S6). En effet, les services représentés sur la figure I.5 représentent des finalités et pas des descriptions de services qui peuvent être différentes pour une même finalité de service. L'initialisation des projets consiste en la description par chaque client de son projet en termes de services disponibles sur le marché comme illustré sur la figure I.5.

Le marché orienté-service met les sous-traitants (fournisseurs de services) au premier plan en obligeant les clients à définir leurs projets en fonction des services existants sur le marché. Plusieurs projets peuvent nécessiter un même service (cas du service S3 sur la figure I.5). Il

est impossible de déclarer un projet sur la place de marché si les services requis pour sa réalisation ne sont pas tous disponibles. Dans l'initialisation de son ou ses projets, le client se limite à la recherche d'abord des services capables de réaliser ses projets sans aucune exigence sur les contraintes de réalisation. Celles-ci seront étudiées dans la phase de négociation de la réalisation du projet entre le client et ses producteurs correspondant aux fournisseurs de services concernés.

6.4 Caractéristiques du marché orienté-service par rapport au marché orienté-projet

Le marché orienté-service se caractérise par une coordination et une collaboration entre les partenaires sans qu'il y ait des exigences imposées à l'avance par un OEM. Dans un marché orienté-projet, les PME à capacités limitées sont dans les derniers rangs de la chaîne logistique établie entre un OEM et ses partenaires (sous-traitants). Un sous-traitant engagé dans la réalisation d'un lot de projet d'un OEM peut définir lui aussi son propre marché orienté-projet en divisant ce lot en sous-lots. Ce mécanisme oblige les PME, ayant des compétences pointues dans un domaine spécifique de la production et de faibles moyens techniques, à se retrouver dans des rangs très bas de la chaîne du fait qu'elles ne peuvent satisfaire que des contraintes faibles.

A la différence des marchés orientés-projets où l'OEM pilote la gestion de ses projets et exige des contraintes de réalisation et des compétences élevées pénalisant une grande partie des PME, le marché orienté-service améliore la présence des PME en réduisant les exigences fortes des OEMs. Cette réduction est due au fait que le marché orienté-service impose le découpage des projets des OEMs en fonction des services existants. Cela permet aux PME à capacités limitées d'avoir l'opportunité d'être présentes dans des rangs plus avancés dans la chaîne de production. La réduction des contraintes exigées par les OEMs offre aux PME des opportunités d'intégration plus équitables. L'intégration de ce type de marché ne dépend plus du respect des contraintes exigées par les OEMs mais plutôt de la possibilité de proposer des services élémentaires répondant aux besoins de ces derniers. Cela améliore la compétitivité du marché où une PME ayant une certaine compétence dans un domaine de la production peut être représentée sur le marché par son service correspondant et peut être ainsi sollicitée par un projet client.

6.5 Difficulté de la gestion des projets multi-sites dans un marché orienté-service

Dans un marché orienté-service, chaque client initie et gère son ou ses projets indépendamment des autres clients. Les producteurs et les clients ne se connaissent pas *a priori*. Suite à une présélection des services qui permettent la réalisation d'un projet pour un client, ce dernier va négocier avec les différents fournisseurs de ces services (les producteurs) la réalisation de son projet. De ce fait, chaque projet client est à l'origine d'un réseau d'entreprises qui regroupe le client et ses différents producteurs, les producteurs des producteurs, etc. En effet, ce réseau d'entreprises prend dans la plupart des cas la forme d'une EV du fait que ce groupement d'entreprises est temporaire et a la même durée de vie que celle du projet. Une fois le projet client réalisé, les collaborations entre les partenaires sont terminées. Aussi, un autre aspect permet de caractériser le réseau d'entreprises créé en tant qu'EV, est l'aspect dynamique. Un producteur peut rejoindre le réseau d'entreprises créé par un projet client du fait de la sollicitation par ce dernier du service fourni par ce producteur, et ce à tout moment du cycle de vie du projet. Aussi, le producteur peut quitter ce réseau à tout instant pour certaines raisons, *i.e.* résultat refusé par le client, etc.

Un service fourni par un fournisseur peut être sollicité par des clients différents du fait que la fonctionnalité de ce service peut être utilisée dans des projets différents (Griffin et Pesch, 2007). Un producteur peut donc participer simultanément à plusieurs EVs comme illustré sur la figure I.6.

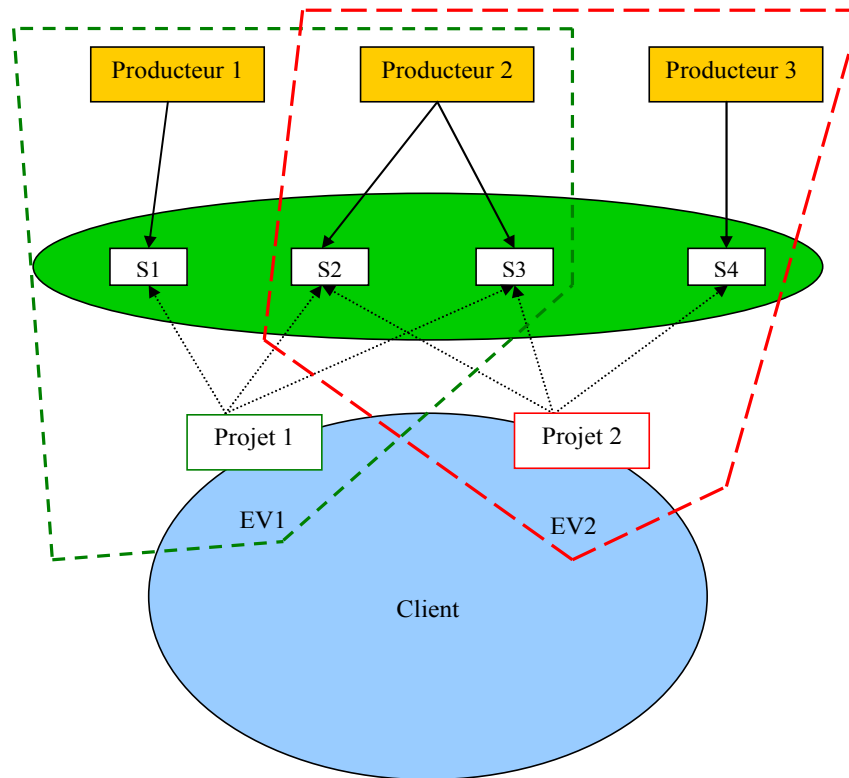


Figure I.6. Participation simultanée d'un producteur à plusieurs EVs

Cette figure montre la participation d'un producteur (producteur 2) à deux EVs différentes (EV1 et EV2) correspondant aux deux projets du client (Projet 1 et Projet 2). Le producteur 2 fournit deux services S2 et S3 demandés par les deux projets client qui sont à l'origine de ces deux EVs.

La présence de multiples EVs dans le marché orienté-service nécessite une gestion distribuée des projets multi-sites faisant intervenir les différents partenaires. Du fait que les fournisseurs de services, *i.e.* les sous-traitants d'un OEM, ne sont pas obligés de respecter les normes et standards utilisés par ce dernier, la distribution de cette gestion met en évidence les problèmes engendrés par l'existence de multiples applications d'entreprise et modèles de gestion de production dans un tel type de marché. Ces problèmes proviennent en fait des caractéristiques de ces applications qui se résument principalement à l'autonomie, à la distribution, et à l'hétérogénéité (Bussler, 2003 ; Hasselbring, 2000). Les applications d'entreprise sont autonomes dans la mesure où elles peuvent être conçues et exécutées indépendamment les unes des autres. La distribution désigne le fait que les applications sont très souvent réparties physiquement sur le réseau d'entreprise. Cette répartition physique est réalisée grâce à une répartition des données et/ou des traitements permettant ainsi l'implémentation au niveau local de certaines données et/ou de certains traitements du système d'information. L'hétérogénéité

est une dimension inhérente au fait que les applications d'entreprise peuvent être développées et déployées de façons indépendantes et selon des approches et des méthodologies différentes.

En fait, les architectures développées à base de différents SCEP organisés en réseau permettent d'appréhender la production multi-site en prenant en compte l'aspect distribué de la gestion des projets dans les marchés orientés-services. Néanmoins, il n'en demeure pas moins que le modèle SCEP n'apporte pas une solution satisfaisante face aux mutations majeures s'opérant dans l'organisation des entreprises sous la pression des nouvelles technologies de l'information et de la communication. Lors de son déploiement dans un environnement multi-site, SCEP présente certaines limites. En effet, ce modèle nécessite une connaissance préalable des sites de production distants représentant les producteurs. Du fait que dans un marché orienté services, les clients et les producteurs ne se connaissent pas à priori, cela nécessite des mécanismes de mise en relation des différents partenaires absents du modèle SCEP. Aussi, le modèle SCEP organisé en réseau permet une communication et une coopération entre des sites qui déploient le même modèle SCEP.

Compte tenu de ces limites, notre objectif peut être résumé par les deux points suivants:

- Localisation des sites distants : afin de permettre au manager de SCEP d'avoir la connaissance des sites avec lesquels il va coopérer ainsi que leurs compétences en termes d'activités de production demandées.
- Interaction indépendante : pour permettre au système qui implémente le modèle SCEP d'interagir avec d'autres systèmes indépendamment des modèles et des plateformes utilisés par ces derniers.

Pour cela, ce modèle doit être complété par des mécanismes de mise en relation des partenaires et de gestion de l'hétérogénéité entre les modèles et méthodes de gestion des projets utilisés sur les différents sites. Le modèle SOA (Service Oriented Architecture) répond bien à cette problématique, illustrant le fonctionnement d'un marché orienté-service et permettant des collaborations dynamiques et des interactions entre les partenaires indépendamment de toute spécificité de certains d'entre eux. Ce modèle est discuté ci-après.

7 Présentation générale et propriétés du modèle SOA

Une architecture orientée service ou SOA (Service Oriented Architecture) rassemble les grandes applications de l'entreprise en services interopérables et réutilisables. En effet, l'architecture orientée service décrit les composants d'un système et la manière de leur interaction à un haut niveau. Les interactions entre les composants sont appelés connecteurs. L'aspect le plus important de SOA est qu'elle sépare l'implémentation du service de son interface.

L'objectif étant d'autoriser les applications ou service à communiquer et à travailler ensemble, quelle que soit leur plate-forme respective, et ce par le biais de services. Un service est un composant indépendant de tout système, dont les interfaces et contrats d'utilisation sont connus. Le service est une notion abstraite pour désigner les fonctionnalités d'un agent logiciel qui implémente une fonctionnalité du service. Un seul service peut avoir plusieurs agents qui l'implémentent (utilisant des langages différents). L'objectif d'une architecture orientée service est de décomposer une fonctionnalité en un ensemble de fonctions basiques, appelées services, fournies par des composants et de décrire finement le schéma d'interaction entre ces services. L'idée sous-jacente est de construire une architecture logicielle globale décomposée en services correspondant aux processus métiers de l'entreprise. Les principaux acteurs qui interviennent dans SOA sont illustrés sur la figure I.7.

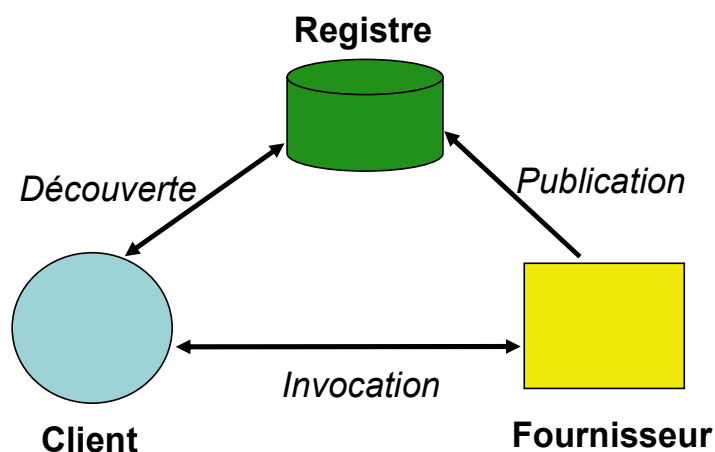


Figure I.7. Principaux acteurs dans SOA

Le fournisseur du service désigne d'un point de vue conceptuel la personne ou l'organisation responsable juridiquement de l'agent logiciel (le service). D'un point de vue opérationnel, les fournisseurs de services peuvent désigner le serveur qui héberge les services déployés. Le client ou consommateur du service représente une personne ou une organisation, cliente

potentielle des services. Du point de vue opérationnel, il désigne l'application cliente qui invoque le service. Le registre ou l'annuaire des services représente la personne ou l'organisation responsable de l'hébergement du dépôt de publication des services. D'un point de vue fonctionnel, il représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services. Le concept registre ou dépôt de service est essentiel dans l'architecture SOA. Il joue un rôle central dans le processus de localisation des besoins, car il est supposé fournir aux clients les informations techniques et sémantiques sur le fonctionnement du service exprimées dans des langages formels (interprétables par les machines). Le service lui-même répondant à quatre caractéristiques : l'indépendance, la découverte, l'invocation dynamique, et l'autonomie. Nickull, (2005) considère qu'il n'existe pas à proprement parler de spécifications officielles d'une architecture SOA, néanmoins les principales notions fédératrices ou concepts clés que l'on retrouve dans une telle architecture sont représentés sur la figure I.8.

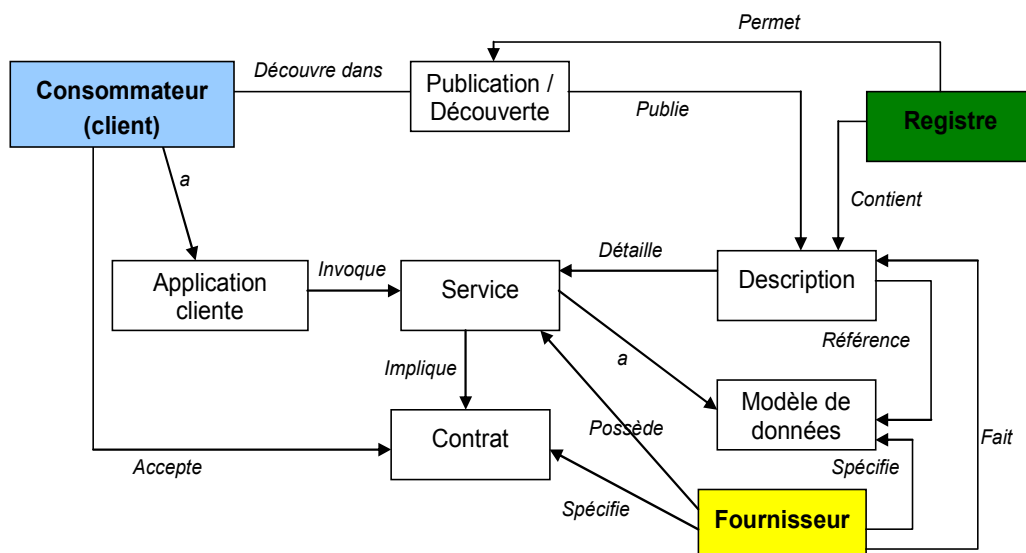


Figure I.8. Concepts clés de SOA et leurs relations (adaptée de (Nickull, 2005))

La notion de service représente une fonction encapsulée dans un composant que l'on peut interroger à l'aide d'une requête composée d'un ou plusieurs paramètres et fournissant une ou plusieurs réponses. Chaque service doit être indépendant des autres afin de garantir sa réutilisabilité et son interopérabilité. La description du service consiste à décrire les paramètres d'entrée du service, le format et le type des données retournées. C'est la spécification du service exprimée dans un langage de description interprétable par les machines. Il existe deux niveaux de description de services. Une description technique dans laquelle le service est vu en termes de messages, de formats, de types, de protocoles de

transport et d'une adresse physique. Cette description joue le rôle d'un contrat d'interaction entre le client et le service. La deuxième description concerne la sémantique du service qui peut avoir une existence formelle ou peut prendre la forme d'un accord entre le fournisseur et le client. La structure des messages qui permettent l'invocation des différentes unités fonctionnelles qui composent le service doit aussi figurer dans la description du service.

La publication consiste à publier dans un registre les services disponibles aux utilisateurs, tandis que la découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés. Le processus de découverte du service doit être capable de répondre aux questions suivantes : où sont localisés les services correspondants ? Et comment les obtenir ? Pour la publication, il existe deux méthodologies : "pull" et "push" (Nickull, 2005). Dans le "pull", le consommateur demande au fournisseur du service de lui envoyer la description du service. Dans le "push", le fournisseur du service ou son agent envoie la description du service aux consommateurs potentiels du service. L'invocation, représentant la connexion et l'interaction du client avec le service. Le contrat de service est une spécification de l'interaction entre le consommateur du service et le fournisseur du service. Il spécifie le format de la requête et de la réponse du service. Il peut aussi spécifier les niveaux de qualité du service. Le modèle de données spécifie les paramètres impliqués dans l'invocation du service.

Dans l'entreprise, SOA contribue à l'évolutivité, à la flexibilité, et à la pérennité du système d'information en encapsulant la complexité de ses applications. Il s'agit de substituer la découpe strictement applicative d'un système d'information par une structuration en services plus réduits et potentiellement plus simple à faire évoluer. Cela offre une plus grande tolérance aux pannes et une maintenance facilitée en réduisant les coûts en phase de maintenance et d'évolution. SOA permet de structurer d'une manière dynamique un système d'information favorisant ainsi son agilité. Ce dynamisme est lié à la possibilité de changer facilement une configuration de services : réutilisabilité possible des composants et principe de couplage faible (ajouter ou enlever un service, modifier l'ordre de communication de services). De plus, SOA offre une localisation et un interfaçage transparent où il est possible d'intervenir sur le contenu de service sans toucher à son interface d'accès (changement interne). Le système d'information est désormais plus facile d'accès, une fois l'exposition des services réalisée, ce qui améliore son accessibilité. SOA simplifie les échanges inter-entreprises et facilite la mise en place à partir d'une application objet existante.

8 Conclusion

Différentes formes de réseaux d'entreprises caractérisent actuellement la production multi-site. Certaines sont de court à moyen terme comme les entreprises virtuelles, d'autres ont une durée de collaboration plus longue comme les entreprises étendues et les chaînes logistiques. Pour gérer leurs collaborations, les entreprises se sont basées sur les technologies d'Internet et ont créé des places de marchés électroniques. Ces marchés sont dans leur majorité des marchés orientés-projets pilotés par les OEMs et dominés par les grandes entreprises, pénalisant un grand nombre de PME ayant des capacités limitées. Comme alternative, nous avons présenté les marchés orientés-services qui améliorent la présence des PME dans les marchés de production.

Dans cet environnement orienté-service, l'hétérogénéité qui peut avoir lieu entre les systèmes et applications de production des partenaires constitue avec l'aspect distribué les principales problématiques dans la gestion des projets de production multi-sites. En effet, la gestion des projets multi-sites nécessite chez les clients des outils qui prennent en compte l'aspect dynamique du marché, la distribution de la décision afin de coopérer avec des partenaires dotés d'outils et méthodes de gestion de production différents. Le modèle multi-agent générique SCEP met en place un mécanisme de gestion distribuée des projets de production multi-sites en faisant coopérer différents sites localisés préalablement et répartis géographiquement. Cependant, ce modèle manque des mécanismes de localisation et de mise en relation des différents sites ainsi que des mécanismes de gestion des hétérogénéités existantes entre les méthodes et modèles de gestion utilisés dans ces différents sites. Le modèle SOA apparaît comme une solution permettant de gérer cette hétérogénéité.

Afin de faire coopérer différents outils et systèmes de production et gérer d'une manière distribuée et indépendante la réalisation des projets multi-sites dans un marché orienté-service, le chapitre suivant fera l'objet de la description et la spécification d'une architecture intégrant les modèles SCEP et SOA pour réaliser cet objectif.

Chapitre II

Architecture interopérable SCEP-SOA pour la gestion de projets multi-sites

1 Introduction

Dans les marchés orientés-services, la réalisation des projets multi-sites nécessite la coopération des différentes applications de production utilisées par les partenaires. Aussi, l'objectif de ce chapitre est de décrire l'architecture SCEP-SOA que nous proposons pour la gestion distribuée de ces projets. Elle est basée sur l'intégration des concepts du modèle multi-agent générique de planification SCEP et des concepts du modèle SOA. L'originalité du modèle de SCEP-SOA provient d'une alliance de la gestion distribuée de projets multi-sites issue du modèle SCEP et de la gestion de l'hétérogénéité des applications apportée par le modèle SOA.

Dans un premier temps, la définition des concepts clés utiles pour la bonne compréhension du modèle sont présentés dans la section 2. La structure des différentes composantes de l'architecture SCEP-SOA proposée est ensuite décrite dans la section 3. Les étapes de fonctionnement de cette architecture comprenant les interactions internes et externes des différentes composantes sont détaillées dans la section 4. Une évaluation illustrant les points forts et les points faibles de cette architecture est ensuite présentée dans la section 5.

2 Définitions des concepts clés dans SCEP-SOA

L'architecture SCEP-SOA proposée est une place de marché orienté-service dédiée à la gestion du cycle de vie d'un produit depuis les spécifications jusqu'à la commercialisation en passant par la fabrication. Elle est basée sur le concept central de service qui est illustré avec les autres concepts sur la figure II.1.

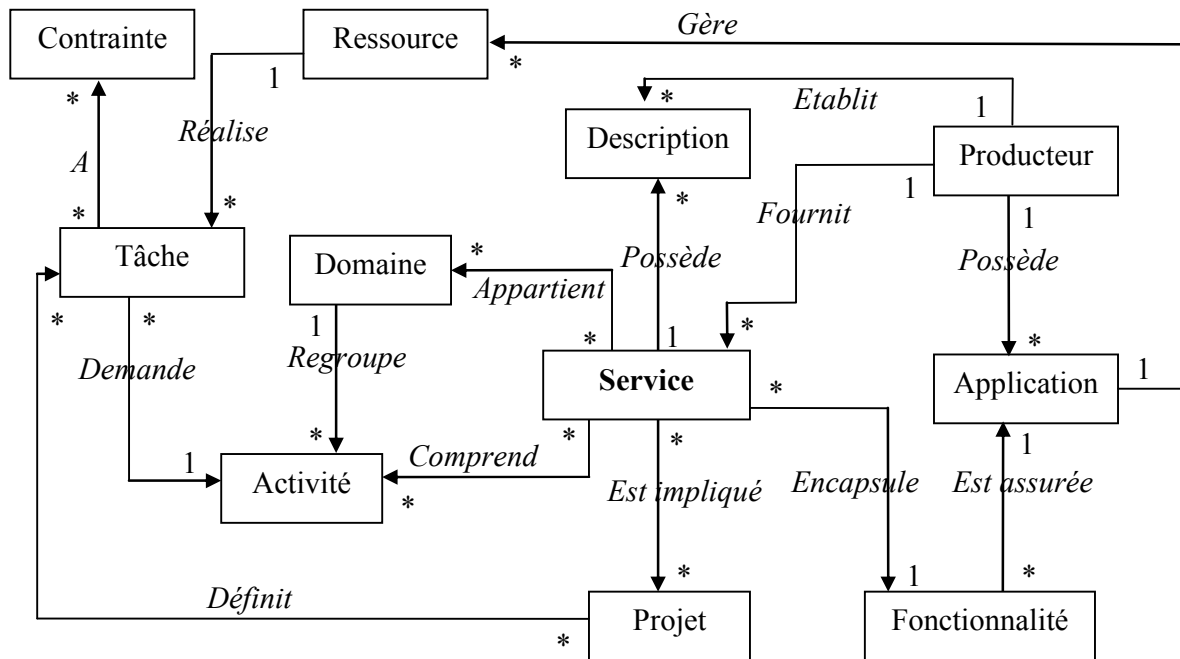


Figure II.1. Concepts clés de SCEP-SOA et leurs relations

- ✓ Un *service* encapsule une *fonctionnalité* précise assurée par une *application*. Il comprend une ou plusieurs *activités* et appartient à un ou plusieurs *domaines*. Il peut être impliqué dans un ou plusieurs *projets*. Un service possède une ou plusieurs *descriptions*. Il est fourni par un seul *producteur*. Cependant, des services fournis par des producteurs différents peuvent avoir la même *fonctionnalité* et concerner les mêmes *activités*.
- ✓ Un *producteur* est l'entité qui possède l'*application*, le *service* encapsulant la *fonctionnalité* assurée par cette *application*, et les *ressources* permettant la réalisation des *activités* fournies par le service. Un producteur peut fournir un ou plusieurs *services*.
- ✓ Une *fonctionnalité* représente une fonction de gestion assurée par une *application*. Une fonctionnalité peut être par exemple la planification, l'ordonnancement, la gestion des achats et des ventes, la gestion comptable, etc.
- ✓ Une *application* représente un outil de gestion de production qui peut avoir une ou plusieurs *fonctionnalités*. Cette application peut être par exemple un ERP, un APS, un CRM, ou un outil de planification, etc.

- ✓ Une *activité* représente un certain savoir-faire dans une des grandes étapes du cycle de vie d'un produit. Par exemple, dans l'étape de la fabrication, une activité peut concerner le fraisage, ou le tournage, etc.
- ✓ Un *domaine* regroupe l'ensemble des *activités* associées à une étape définie. Soit x une étape du cycle de vie du produit. Dans la suite de ce document, nous désignons, par abus de langage, « domaine de x » l'ensemble des activités associées à l'étape x . Par exemple, le domaine de fabrication regroupe l'ensemble des activités de fabrication, *i.e.* tournage, perçage, fraisage, etc.
- ✓ Une *ressource* représente l'outil physique géré par une *application* et qui permet la réalisation d'une ou de plusieurs *tâches*. Par exemple, une ressource de fabrication peut être une fraiseuse, ou une perceuse, etc.
- ✓ Une *tâche* concerne la réalisation d'une *activité* sur une *ressource* pour une durée déterminée. Elle possède un ensemble de *contraintes*, *i.e.* contraintes temporelles comme date de début, date de fin, etc. Une seule valeur est possible pour une contrainte. Notons que plusieurs *tâches* peuvent être définies pour une même *activité*.
- ✓ Un *projet* définit un ensemble de *tâches* pour une ou plusieurs *activités* fournies par un ou plusieurs *services*, dont la réalisation permet d'avoir un produit fini. Un projet est considéré réalisable si l'ensemble des *activités* demandées par ce projet sont fournies par des *services* existants.
- ✓ La *description* d'un service est réalisée par le producteur. Elle contient des propriétés fonctionnelles et non fonctionnelles. Les propriétés fonctionnelles concernant l'aspect technique du service, *i.e.* paramètres d'entrée et de sortie, pré-conditions, post-conditions, adresse d'invocation, etc. Les activités fournies par le service font partie des paramètres d'entrée et de sortie. La description du service comprend aussi parmi les paramètres d'entrée un paramètre de confirmation noté *customer_confirmation*. Ce paramètre est défini pour chaque activité du service. Il servira dans la confirmation des résultats établis par ce producteur chez les clients qui vont invoquer le service. Un paramètre de sortie noté *producer_confirmation* est aussi introduit pour permettre au producteur de confirmer au client la prise en compte de la confirmation du résultat établi. L'utilisation de ces deux paramètres est détaillée dans les sections suivantes.

Les propriétés non fonctionnelles concernent l'aspect non technique du service, *i.e.* nom du fournisseur, coordonnées du fournisseur, description générale du service, etc. Elle contient aussi le contrat de service spécifiant les niveaux de la qualité du service offert ainsi que la durée maximale d'attente pour la réponse du service. L'expiration de cette durée, pour n'importe quelle raison, indique l'échec de l'invocation courante du service et nécessite par la suite une ré-invocation de ce service. Un service peut posséder une ou plusieurs *descriptions*.

3 Composantes de l'architecture SCEP-SOA

L'architecture SCEP-SOA proposée s'articule autour des trois acteurs principaux du modèle SOA : le registre de services contenant la description des services, le producteur ou fournisseur de service qui possède le service, et le client ou consommateur du service qui utilise le service.

3.1 Structure du registre de services

Le registre de services est l'entité contenant la description des services fournis par les différents producteurs. Il permet la mise en relation du client et des producteurs. Le propriétaire du registre est l'entreprise implémentant le registre. Elle est chargée de son contrôle. La structure d'un registre de service est représentée sur la figure II.2.

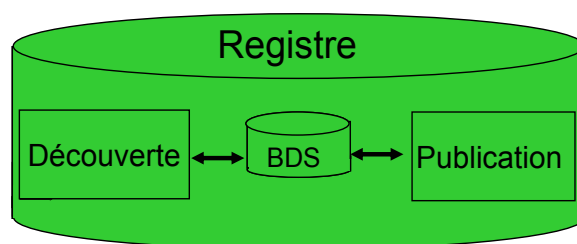


Figure II.2. Registre de services dans SCEP-SOA

Le registre contient une interface de communication permettant aux clients de découvrir les services conduisant à la réalisation de leurs projets et aux producteurs de déclarer leur savoir-faire en publiant leurs services. Le service de découverte permet aux clients de rechercher les services ainsi que leurs producteurs. Le service de publication permet aux producteurs de publier la description de leurs services. Ces deux services communiquent avec une base de données de services (BDS) contenant des informations sur les services fournis et leurs fournisseurs, *i.e.* les propriétés fonctionnelles et non fonctionnelles. Dans la BDS, les services

sont stockés par domaine. Les domaines sont définis et implémentés par l'administrateur du registre.

3.2 Structure du producteur

Un producteur est une entité qui possède des savoir-faire représentés sous forme d'activités de production, réalisables par un ensemble de ressources locales gérées par une application interne de gestion. La structure d'un producteur est donnée sur la figure II.3.

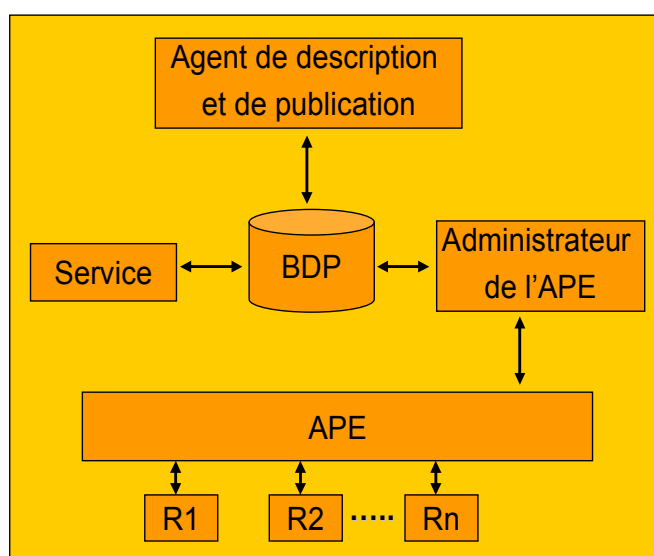


Figure II.3. Producteur dans SCEP-SOA

Chaque producteur possède une application de production notée APE (Application de Production de l'Entreprise), une base de données notée BDP, un agent de description et de publication, le(s) service(s) fournis, et des ressources (R1, R2, etc.) permettant la réalisation des activités fournies par le(s) service(s).

L'exposition des activités au public se fait via des services qui implémentent les fonctions de production assurées par l'APE. Ces services sont décrits dans la base de données locale BDP stockant la description des services, les informations concernant la description des projets clients en fonction des activités fournies par les services, et les résultats de gestion établis par l'APE pour ces projets. Un résultat correspond par exemple à un plan d'ordonnancement pour un service de planification.

Dans notre approche, les services considérés ici sont des services à états (statefull services) car ils conservent l'état des invocations précédentes. En fait, une invocation d'un service peut prendre en compte le résultat établi par une invocation précédente en confirmant la totalité ou

une partie de ce résultat. L'état de l'invocation concerne l'état des projets concernés et des résultats correspondants existant dans la BDP. Un projet existant peut avoir trois états : non traité, traité, ou validé. L'état *non traité* signifie que le projet doit être traité dans le prochain lancement de l'APE. L'état *traité* signifie qu'un résultat a été établi pour ce projet mais n'est pas encore validé par le client. L'état *validé* signifie que le projet est validé par le client correspondant. Un résultat existant pour un projet peut avoir deux états : non validé, ou validé. Un résultat *non validé* correspond à un projet *traité*. Un résultat *validé* correspond à un projet *validé*. La gestion de ces états est détaillée dans la suite.

Les services sans états (stateless services) ne conviennent pas à notre cas car ils ne conservent pas l'état des invocations précédentes. Cela provient du fait que pour une validation d'un résultat d'une invocation du service auprès d'un producteur, un client doit invoquer en permanence le service chez ce producteur.

L'administrateur de l'APE interagit avec la BDP pour récupérer les projets issues des clients afin de les traiter, et stocke dans cette BDP les résultats établis pour ces demandes dans chaque cycle de gestion. Un cycle de gestion correspond à un lancement de l'APE. La durée d'un cycle de gestion est appelée fréquence de lancement de l'APE. Elle est égale au temps d'attente avant le lancement de l'APE plus le temps de lancement de l'APE et l'enregistrement des résultats dans la BDP.

En effet, plusieurs stratégies peuvent être appliquées pour définir la fréquence de lancement de l'APE. Ces stratégies sont inspirées des travaux présentés dans (Kart *et al.*, 2006). Une stratégie peut être définie par un seuil de projets qui représente un nombre maximal de projets cumulés. Quand ce seuil est atteint, l'administrateur de l'APE récupère les projets enregistrés et les charge dans l'APE pour les traiter.

D'une autre manière, une stratégie peut être basée sur un « timeout » fixant un temps pendant lequel les projets sont cumulés. Toute demande de gestion d'un projet venant à la fin d'un « timeout » sera traitée dans le prochain cycle de gestion. Une stratégie peut être aussi définie en prenant en compte le critère de temps d'attente maximal. Selon cette stratégie, à chaque réception d'une demande, un compteur de temps d'attente se déclenche. Quand le compteur d'une demande atteint le temps maximal, le traitement est effectué.

Une stratégie peut se baser aussi sur une combinaison du « timeout » et d'un seuil de projets, ou sur le choix effectué par l'administrateur de l'APE qui décide à un certain moment de lancer l'application. Dans le premier cas, si une demande est reçue quand le seuil dans la BDP est atteint même avant la fin du timeout, cette demande ne sera pas traitée par le producteur. Cela conduit à l'expiration du temps d'attente maximale pour le résultat de l'invocation du service chez le client indiquant à celui-ci que sa demande n'a pas été traitée. A noter que le producteur n'est pas obligé de préciser ce nombre dans la description du service.

Dans l'architecture SCEP-SOA proposée, nous nous sommes basés sur la stratégie qui définit un « timeout » au bout duquel l'APE sera lancée. L'intérêt de cette stratégie est de ne pas faire attendre, pour une durée inconnue, les clients qui ont invoqué le service. Du fait de l'utilisation des services à état, le producteur doit garder dans sa BDP les résultats établis par l'APE pour une certaine durée en attendant la confirmation de la part du client ayant invoqué le service.

L'agent de description et de publication comprend un module de description des services dans la BDP, et un module de publication de cette description dans le registre des services.

Les interactions entre ces différents éléments sont détaillées avec le fonctionnement de l'architecture présenté dans la section 3.4.

3.3 Structure du client

Le client souhaite réaliser certains projets, définis en fonction des services découverts dans le registre. La structure du client intégrant le modèle SCEP est représentée sur la figure II.4.

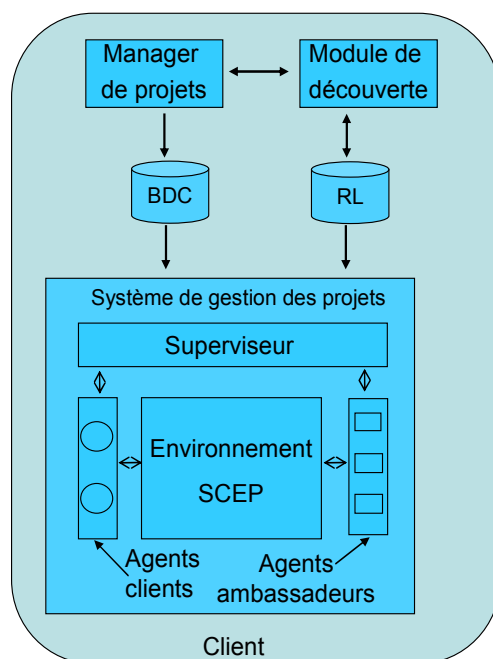


Figure II.4. Client intégrant le modèle SCEP

Le client dispose d'un manager de projets responsable de la réalisation de ses projets, d'une base de données BDC contenant la description des projets définis par le manager de projets, d'un module de découverte permettant de découvrir les services recherchés ainsi que leurs producteurs, d'un registre local RL, et d'un système de gestion de projets.

Le registre local est une copie limitée du registre global, qui stocke des informations sur les services les plus fréquemment sollicités par le client ainsi que les producteurs avec lesquels il a eu de bonnes collaborations. Il permet d'accélérer la découverte des services, de mieux guider le client dans le choix de ses partenaires en privilégiant les producteurs les plus sollicités, améliorant ainsi les relations de partenariat entre le client et les producteurs avec lesquels il a déjà collaboré. Localement, le module de découverte recherche dans le registre local (RL) les services permettant la réalisation des projets. Il peut rechercher également dans le registre global pour découvrir les services non trouvés dans le RL.

Un coefficient de collaboration est défini dans le registre local pour chaque producteur. Il est incrémenté par le manager des projets clients après chaque collaboration avec le producteur correspondant. Ce coefficient a pour objectif de privilégier les meilleurs producteurs pour une même activité, de garantir une meilleure collaboration, et d'entretenir une relation de partenariat à long terme. La cohérence entre les informations contenues dans le registre local et celles contenues dans le registre global doit être vérifiée régulièrement afin de prendre en

compte les modifications effectuées dans ce dernier, *i.e.* suppression de certains producteurs du registre global, modification de la description de certains services, etc.

Le système de gestion des projets permet de gérer d'une manière distribuée la réalisation des projets clients par collaboration avec les producteurs sélectionnés. Du fait que le modèle SCEP est bien adapté à cet objectif, les agents de ce modèle sont intégrés dans le système de gestion des projets pour gérer d'une manière distribuée et autonome la réalisation des projets clients. La gestion des projets s'effectue d'une part par coopération entre les agents clients représentant les projets, et les agents ambassadeurs représentant les producteurs, via l'environnement SCEP, et d'autre part par invocation des services chez les producteurs par les agents ambassadeurs correspondants.

3.4 Interactions entre les acteurs dans SCEP-SOA

Dans SCEP-SOA, les interactions entre les différents acteurs sont représentées sur la figure II.5.

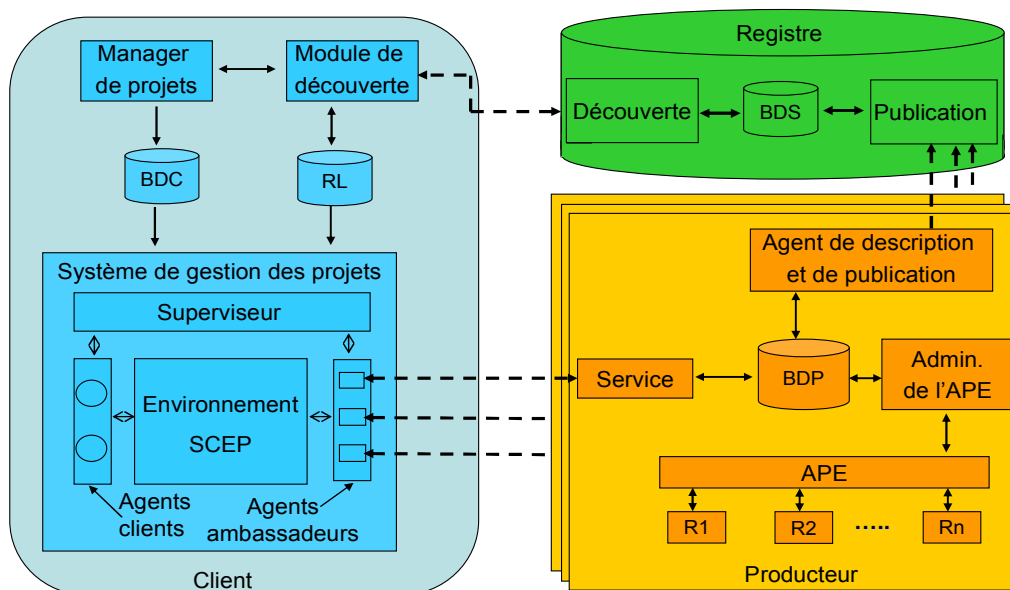


Figure II.5. Interactions entre les acteurs dans SCEP-SOA

En effet, ces interactions sont issues des interactions dans l'architecture SOA présentée précédemment et concernent deux grandes étapes : une étape de mise en relation des partenaires, et une étape d'invocation des services et gestion des projets clients.

La première étape se résume par la publication des services par les producteurs et la découverte de ces services par les clients intéressés. Dans la publication, les acteurs concernés

sont le registre et le producteur publiant les descriptions des services fournis dans le registre. La découverte des services fait intervenir et le registre et le client qui peut ainsi découvrir les services publiés et leurs producteurs.

La deuxième étape met en interaction le client et ses producteurs. Cette étape comprend l'initialisation et la mise en place du système de gestion de projets intégrant le modèle SCEP et l'invocation des services distants fournis par les producteurs en faisant intervenir les agents ambassadeurs du modèle SCEP chez le client (demandeurs de services) et les services des producteurs concernés.

4 Fonctionnement de l'architecture SCEP-SOA

Le fonctionnement global de l'architecture SCEP-SOA est décrit par le diagramme de cas d'utilisation de la figure II.6.

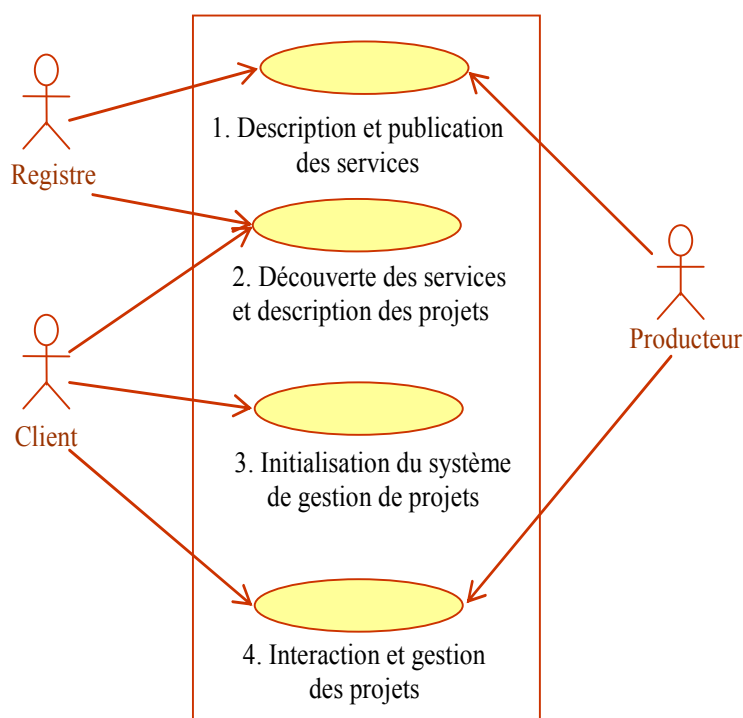


Figure II.6. Diagramme de cas d'utilisation de SCEP-SOA

Ces cas d'utilisation correspondent aux interactions présentées précédemment. Le premier cas concerne la description et la publication des services par le producteur. Le deuxième représente la découverte des services et la description des projets par le client. Le troisième concerne l'initialisation et la mise en place du système de gestion de projets. Enfin, le

quatrième cas concerne l'interaction et la négociation entre le client et les producteurs concernés pour la gestion de la réalisation des projets clients.

4.1 Description et publication des services

Le diagramme de séquences de la description et de la publication des services par un producteur est illustré sur la figure II.7.

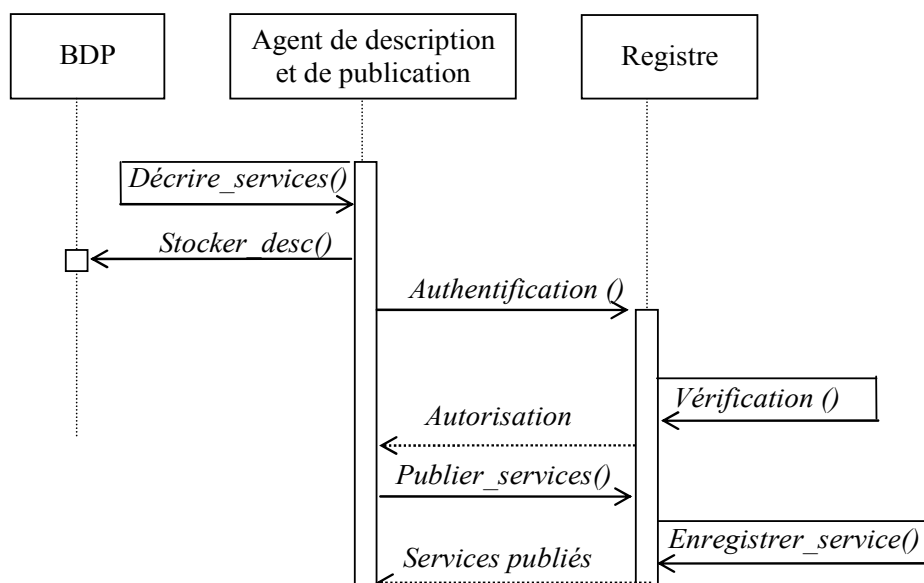


Figure II.7. Description et publication des services

L'agent de description et de publication décrit les services fournis par le producteur et stocke ces descriptions dans la base de données (BDP). Pour les publier, cet agent invoque le service de publication du registre. Il s'authentifie d'abord auprès du registre s'il est déjà inscrit, dans le cas contraire, il doit demander la création de son compte auprès de l'administrateur du registre. Après l'authentification de l'agent de description et de publication du producteur, ce dernier peut publier ses services dans les domaines concernés, voir les services qu'il a offerts, modifier les descriptions de ses services, ou publier de nouveaux services.

A chaque réception d'une nouvelle description de service, le registre enregistre dans sa base de données les informations contenues dans la description. Après l'enregistrement du service, le registre retourne au producteur correspondant une confirmation de la publication de ses services.

Afin de diminuer la charge du registre, les domaines de services peuvent être distribués sur plusieurs registres.

4.2 Découverte des services et description des projets clients

L'objectif de la découverte des services est de permettre aux clients désirant réaliser certains projets d'identifier les services permettant la réalisation des activités concernées. Cette découverte est réalisée par le module de découverte des services, et illustré par le diagramme de séquences de la figure II.8.

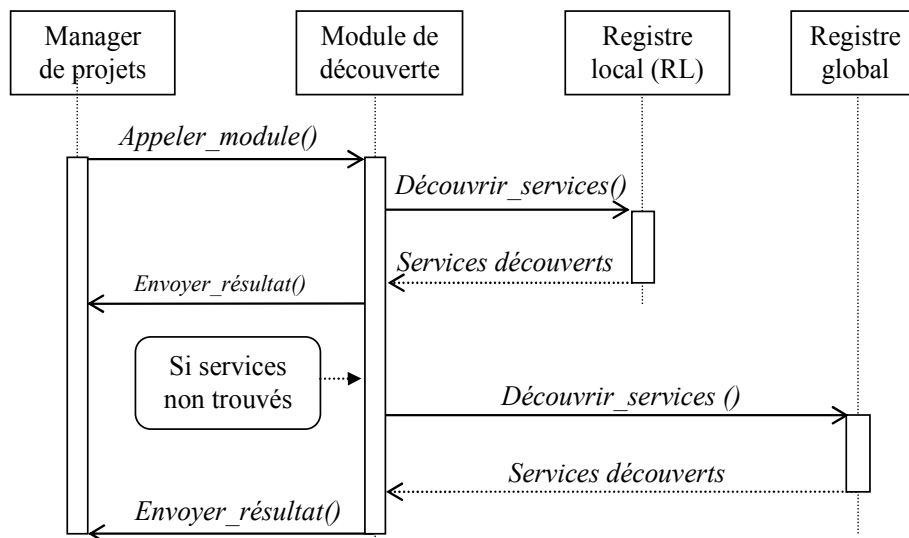


Figure II.8. Découverte des services et description des projets

Suite à l'appel du module de découverte des services par le manager de projets, ce module recherche d'abord dans le registre local (RL) afin de trouver les services les plus fréquemment sollicités par le client. Dans la demande de découverte, le manager de projets doit préciser le(s) domaine(s) des services recherchés ainsi que les activités souhaitées.

Dans le cas où certains services ne sont pas trouvés dans le registre local, le module de découverte de services du client invoque le service de découverte du registre global. Les résultats de la découverte, locale ou globale, sont transmis ensuite au manager de projets. La découverte nécessite des mécanismes de rapprochement sémantique afin d'établir les correspondances entre les activités recherchées par le client et celles définies par les services publiés dans le registre global.

Suite à la découverte des services, le manager de projets alimente le registre local (RL) avec l'ensemble des services sélectionnés ainsi que leurs producteurs, et la base de données (BDC) avec les contraintes souhaitées pour les activités réalisées par ces services.

4.3 Initialisation du système de gestion de projet chez le client

Une fois les projets décrits et stockés dans la base de données du client (BDC), le manager de projets le signale à l'agent superviseur du système de gestion de projets.

Après la récupération de la description des projets auprès de la BDC et des descriptions des services découverts auprès du registre local RL, le superviseur crée les agents clients et ambassadeurs, et initialise l'environnement SCEP. Cet environnement est une place de marché orientée-service dans laquelle les agents ambassadeurs publient les résultats de l'invocation des services émanant des producteurs correspondants, et les agents clients découvrent les résultats correspondant à la gestion des projets qu'ils gèrent.

En effet, les informations récupérées par le superviseur sont réparties en trois ensembles : l'ensemble des services découverts, l'ensemble des activités fournies par ces services, et l'ensemble des producteurs correspondants. Dans l'environnement SCEP, le superviseur établit une correspondance entre ces trois ensembles avec les ensembles des agents clients et ambassadeurs, comme illustré sur la figure II.9. Deux agents ambassadeurs, trois agents clients, sept activités et cinq services y sont représentés. Chaque agent client (ACi) peut être lié à une ou plusieurs activités (At).

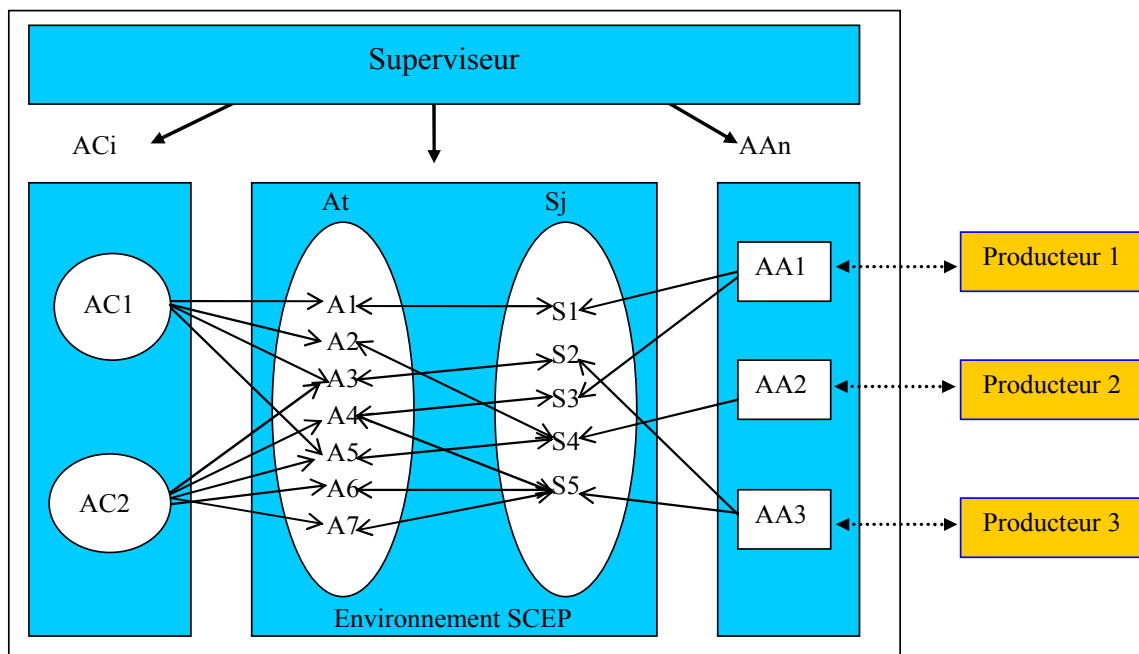


Figure II.9. Système de gestion de projets basé sur SCEP

Une activité peut être liée à un ou plusieurs agents clients (par exemple l'activité A5) ainsi qu'à un ou plusieurs services (Sj), illustrant ainsi la sollicitation d'un même service (par

exemple S4) par plusieurs projets clients. Chaque service peut être lié à une ou plusieurs activités. Chaque agent ambassadeur (AAn), représentant un producteur, peut être lié à un ou plusieurs services illustrant le fait qu'un producteur peut fournir plusieurs services. Chaque service peut être lié à un seul agent ambassadeur du fait qu'un service est fourni par un seul producteur. La vue d'ensembles présentée sur la figure II.9 est un exemple illustratif. En effet, dans l'environnement SCEP, les services ne figurent qu'à travers les activités qu'ils fournissent. Ils n'existent pas dans l'environnement et leur connaissance est propre aux agents ambassadeurs correspondants.

La connaissance d'un agent client se limite à l'ensemble des activités fournies par les services découverts pour la réalisation du projet qu'il représente. Il n'a aucune connaissance des producteurs ni des services. D'autre part, chaque agent ambassadeur a une connaissance sur les activités fournies par les services du producteur qu'il représente, il n'a pas de connaissance des autres services ni de la composition des projets clients.

Suite à la création de l'environnement SCEP et de ses agents, chaque agent client définit, d'une manière indépendante des autres agents clients, son projet en fonction des activités des services découverts en précisant pour chaque activité les contraintes pour les tâches associées. Un exemple de description des activités dans l'environnement SCEP est présenté sur la figure II.10.

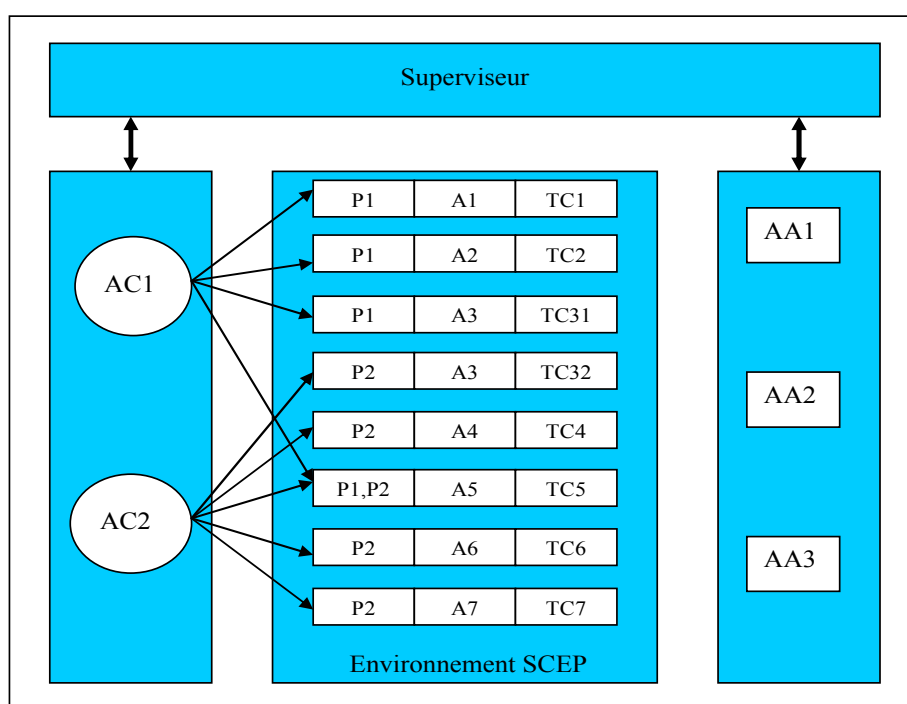


Figure II.10. Initialisation des activités des projets clients dans l'environnement SCEP

Nous considérons l'exemple présenté précédemment sur la figure II.9. Deux agents clients sont représentés (AC1 et AC2) pour les des deux projets clients P1 et P2 respectivement. L'AC1 est responsable du projet P1 qui demande les activités A1, A2, A3 et A5. L'AC2 est responsable du projet P2 qui demande les activités A3, A4, A5, A6, et A7. A chaque activité d'un projet est associé un ensemble de tâches avec les contraintes correspondantes (TC). Par exemple, TC1 représente l'ensemble des tâches du projet P1 demandant l'activité A1 ainsi que leurs contraintes correspondantes. Une activité sollicitée par plusieurs projets peut avoir le même ensemble TC (cas de l'activité A5), ou des ensembles différents (cas de l'activité A3).

4.4 Interaction et gestion des projets

Une fois l'initialisation de l'environnement SCEP et de ses différents agents faite par le superviseur, des interactions sont déclenchées entre ces agents ainsi qu'avec les producteurs afin de gérer et négocier la réalisation des projets. Dans la gestion des projets, deux types d'interaction sont identifiés : les interactions internes et les interactions externes. Les interactions internes représentent les interactions entre les éléments d'une même composante, par exemple entre les agents clients et les agents ambassadeurs chez le client. Les interactions externes représentent les interactions entre des éléments appartenant à des composantes différentes, par exemple entre les agents ambassadeurs du client et les services des producteurs correspondants.

Nous identifions trois étapes durant la gestion d'un projet : collecte et transmission des contraintes de projets aux producteurs, traitement des projets par les producteurs et envoi des résultats au client, traitement et validation des résultats par le client. Chez le client, la première et la deuxième étape nécessitent des interactions externes ; la dernière étape met en place des interactions internes.

Le choix de la granularité des services et la stratégie de leur invocation ainsi que les différentes étapes d'interaction sont détaillés dans ce qui suit.

4.4.1 Choix de la granularité du service

La granularité d'un service est la quantité totale de données encapsulée par ce service (SOA Glossary).

Si la granularité est très faible, par exemple de l'ordre d'une activité par invocation, la complexité de l'algorithme de gestion augmente du fait que le client doit effectuer un nombre important d'invocation de services pour gérer ses projets. Par conséquent, le risque de surcharge du réseau de communication établie entre le client et les producteurs augmente. Aussi, en choisissant une telle granularité, les contraintes qui peuvent exister entre les activités d'un projet, *i.e.* liens temporels, ne sont pas respectées du fait que les activités d'un projet peuvent être traitées dans des cycles de gestion différents chez les producteurs. Cela ralentit la convergence de l'algorithme qui représente l'obtention d'un résultat de gestion qui soit validé par le client.

Pour une granularité de l'ordre d'un projet par invocation, les contraintes entre les activités sont bien respectées du fait que les informations concernant toutes les activités d'un projet sont contenues dans la requête d'invocation du service. Le nombre d'invocation de service diminue. Cependant, du fait qu'un cycle de gestion chez un client nécessite l'obtention d'un résultat de gestion pour chaque projet client défini, la convergence de l'algorithme reste lente due à la possibilité que les projets peuvent être traités dans des cycles de gestion différents chez les producteurs.

Bien que la quantité de données soit importante, nous considérons une granularité de service de l'ordre de l'ensemble des projets à gérer. La requête d'invocation du service contient des informations concernant tous les projets clients demandant des activités fournies par ce service. Cette granularité garantit le respect des contraintes entre les activités d'un projet et diminue le nombre d'invocations de service qui est de l'ordre d'une invocation par cycle de gestion. Cela évite que certains projets ou certaines activités demandées par un projet soient traitées dans des cycles de gestion différents chez les producteurs du fait que dès que le service est invoqué, toutes les informations concernant les projets ont été transmises au producteur correspondant. Avec cette granularité, la convergence de l'algorithme de gestion est plus rapide que dans les autres cas. La convergence de la méthode a été démontrée dans (Archimède et Coudert, 2001b).

4.4.2 Stratégie d'invocation des services

En effet, les agents ambassadeurs n'invoquent pas en même temps les services chez les producteurs. Cela provient du fait que les producteurs ne lancent pas leur application de gestion (APE) au même moment. Rappelons que chez le client, un cycle de gestion

correspond à l'obtention d'un résultat de gestion (plan d'ordonnancement par exemple) pour tous les projets existant dans l'environnement SCEP. La durée du cycle de gestion chez le client est égale à la fréquence maximale de lancement de l'APE chez les producteurs concernés par le cycle de gestion courant. Les durées de ces cycles peuvent varier. Par exemple, considérons un client qui interagit avec deux producteurs pour la gestion de son projet. Le premier producteur lance son APE toutes les 10 minutes et le deuxième toutes les 20 minutes. S1 représente le service du premier producteur et S2 le service fourni par le deuxième.

Au début de la gestion, les deux producteurs sont concernés. Donc, les invocations des services doivent être ordonnées dans l'ordre décroissant par rapport à la fréquence de lancement. En d'autres termes, S2 doit être invoqué d'abord, et après 10 minutes (la différence de temps entre les fréquences de lancement), S1 est invoqué. Cela permet, dans le cas du délai d'attente maximale pour les deux services, de recevoir les deux résultats d'invocation en même temps, permettant ainsi au client de traiter ces résultats ensemble. Dans ce cas, la durée du cycle de gestion du client est égale à 20 minutes représentant la fréquence maximale de lancement de l'APE par les producteurs concernés.

Si le client valide le résultat de S1 mais pas celui de S2, alors, dans le prochain cycle de gestion, le client invoque seulement le service S2. Dans ce cas, la durée du cycle de gestion est égale à 10 minutes, qui est la fréquence de lancement maximale pour les producteurs concernés. Si tous les résultats d'invocation des services sont reçus par les agents ambassadeurs avant la fin de la durée d'un cycle de gestion, le cycle prend fin et le client passe ensuite à l'étape de traitement et validation des résultats reçus.

Cependant, si on considère que les deux services sont invoqués simultanément, dans le cas d'attente maximale pour les réponses des services, le client reçoit le résultat de S2 et après 10 minutes celui de S1. Cela nécessite l'attente du résultat de S2 pour 10 minutes avant d'être traité par le client. Cela peut conduire à l'expiration de la durée de vie de ce résultat chez le producteur 2 qui définit une durée de stockage du résultat dans la BDP. Normalement, cette durée est égale à la fréquence de lancement de l'APE. Cela conduit à l'impossibilité de validation de ce résultat par le client. Ce problème est illustré sur la figure II.11 en négligeant tous les délais autres que la durée du cycle de gestion, *i.e.* délai de transmission de l'invocation au producteur, délai de transmission du résultat de l'invocation au client, délai du traitement et de l'établissement du résultat chez le producteur, etc.

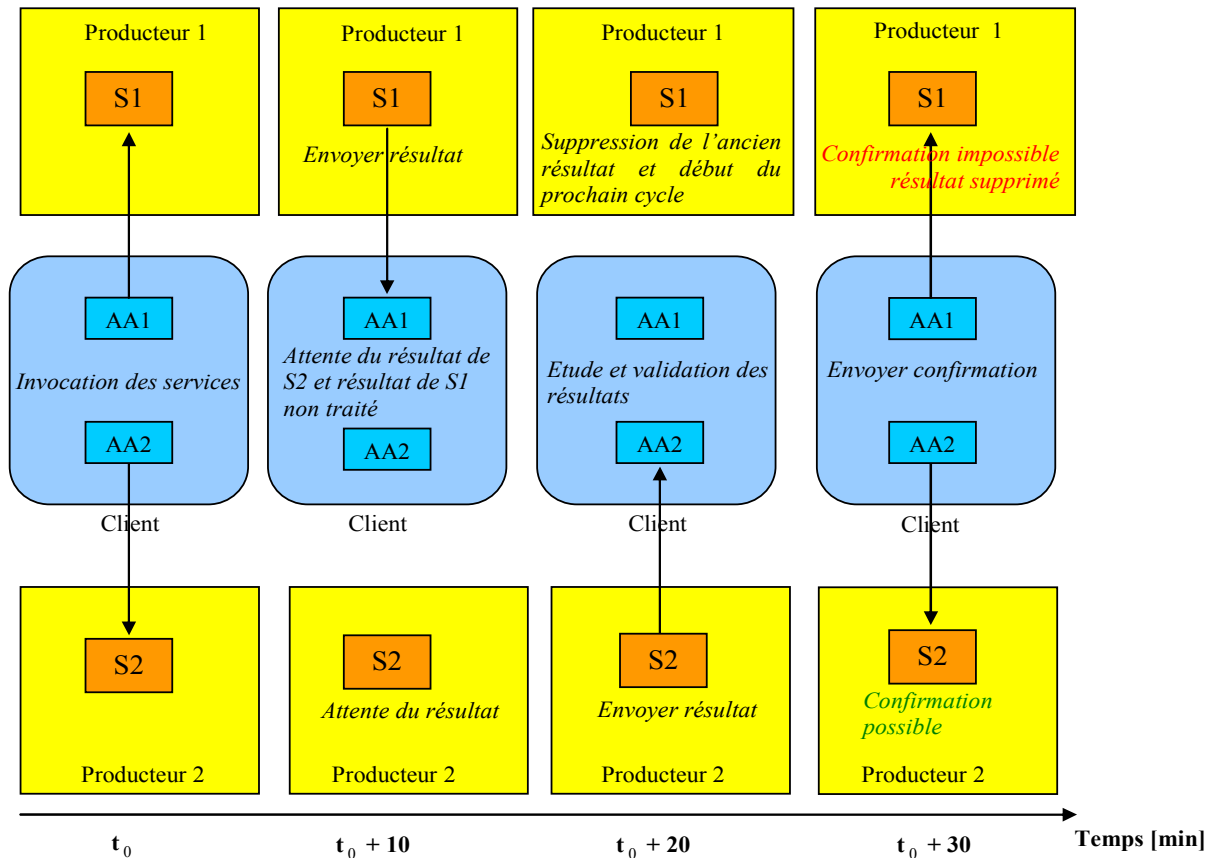


Figure II.11. Problème de l'invocation simultanée des services chez deux producteurs

L'exemple présenté montre bien l'impossibilité de confirmation d'un résultat qui a été supprimé chez le producteur (temps $t+30mn$). En effet, l'application de cette stratégie favorise le choix du producteur ayant la fréquence de lancement la plus longue au détriment des autres producteurs. Cela ne convenant pas pour les marchés orientés-services qui cherchent à améliorer la présence des PME indépendamment de la fréquence de lancement de leur application de gestion de production et des autres particularités.

4.4.3 Collecte et transmission des contraintes de projets aux producteurs

Le diagramme de séquences pour la collecte des tâches et des contraintes par un agent ambassadeur et la transmission de ces informations au producteur correspondant est illustré sur la figure II.12.

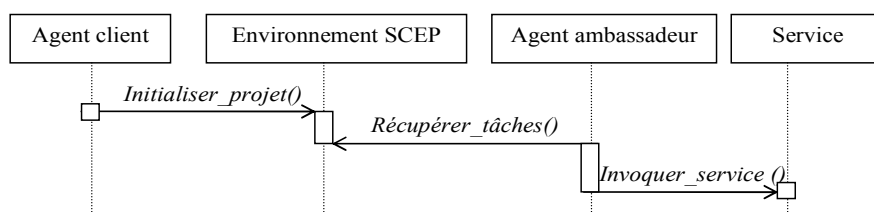


Figure II.12. Collecte et transmission des contraintes au producteur

Une fois l'initialisation des projets faite par les agents clients dans l'environnement SCEP, suite à l'invitation du superviseur, chaque agent ambassadeur vient dans cet environnement récupérer l'ensemble des tâches et de leurs contraintes (TC) pour les activités concernées par les services du producteur qu'il représente. La figure II.13 illustre la collecte des informations par les agents ambassadeurs pour l'exemple décrit sur la figure II.9.

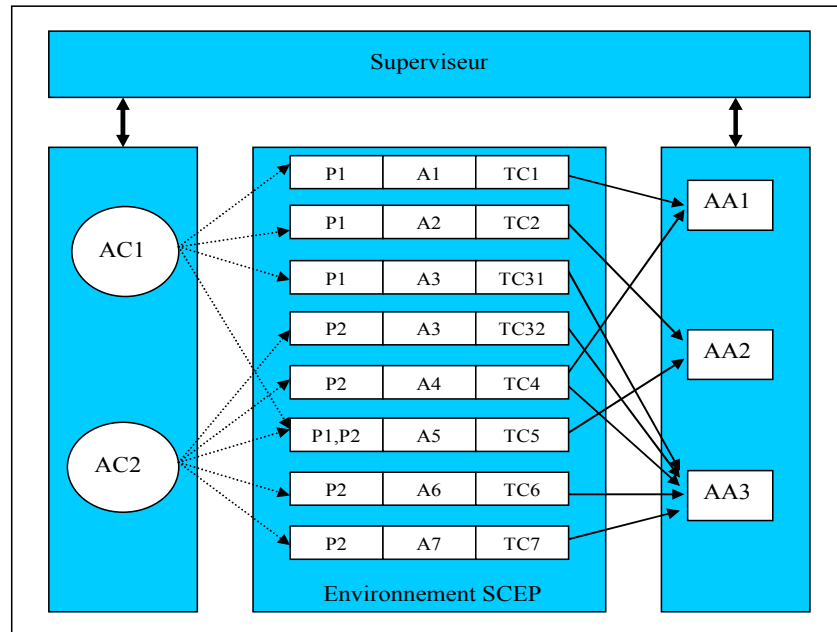


Figure II.13. Récupération des contraintes de l'environnement SCEP par les agents ambassadeurs

Dans cet exemple, les services présents dans l'environnement SCEP sont fournis par trois producteurs représentés respectivement par les agents ambassadeurs AA1, AA2 et AA3. Chaque agent ambassadeur récupère de l'environnement SCEP l'ensemble TC des tâches et des contraintes souhaitées, *i.e.* contraintes temporelles, contraintes de coût de réalisation, etc. pour les activités fournies par les services du producteur correspondant. L'agent ambassadeur AA1 est concerné par les activités A1 et A4 contenues dans les services fournis par le producteur qu'il représente (à savoir les services S1 et S3). L'agent ambassadeur AA2 est concerné par les activités A2 et A5 contenues dans le service fourni par le producteur qu'il représente (à savoir le service S4). L'agent ambassadeur AA3 est concerné par les activités A3, A4, A6, et A7 contenues dans les services fournis par le producteur qu'il représente (à savoir les services S2 et S5).

Pour chaque ensemble TC récupéré, l'agent ambassadeur note l'activité correspondante ainsi que les projets concernés afin d'introduire ces informations dans la requête d'invocation du

service correspondant. Ces informations vont figurer aussi dans les résultats de l'invocation des services et serviront dans la découverte de ces résultats par les agents clients.

Après la collecte de ces informations, chaque agent ambassadeur invoque le(s) service(s), chez le producteur correspondant. L'invocation du service concerne l'envoi par l'agent ambassadeur d'une requête contenant l'ensemble des tâches et leurs contraintes pour les activités contenues dans le service au producteur correspondant. Cette requête contient aussi les identifiants des projets clients concernés par cette invocation.

Pour une première invocation du service, le paramètre de confirmation *customer_confirmation* est initialisé à « Non » pour chaque activité demandée. Cela permet au producteur concerné de savoir qu'il ne s'agit pas d'une validation d'un résultat déjà établi, mais plutôt d'une nouvelle invocation.

4.4.4 Traitement des projets par les producteurs et envoi des résultats au client

Le diagramme de séquences du traitement d'une invocation du service chez un producteur et la proposition d'un résultat est illustré sur la figure II.14.

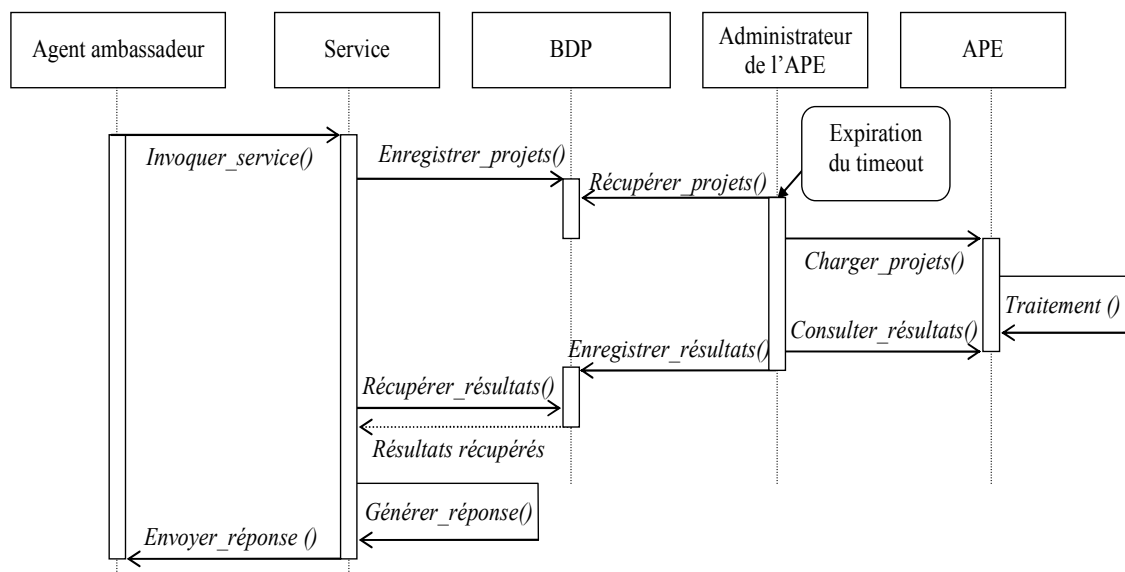


Figure II.14. Invocation du service et envoi du résultat

Une fois la requête d'invocation émanant d'un agent ambassadeur reçu par le service, ce dernier stocke les informations contenues dans cette requête dans la BDP en identifiant les différents projets à gérer.

La procédure d'enregistrement de projets dans la BDP nécessite une vérification par le service de l'état de l'invocation en cours. Le diagramme d'états-transitions d'un service chez un producteur est représenté sur la figure II.15. Il représente les états des projets clients demandant ce service et des résultats établis pour ces projets chez un producteur.

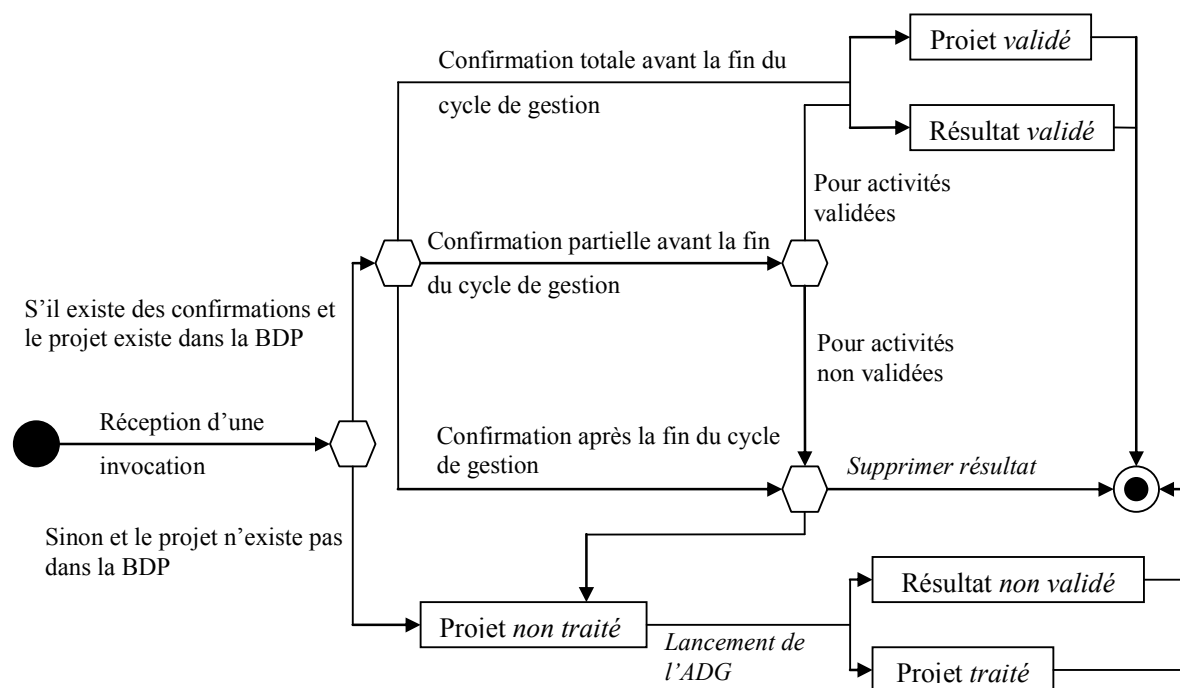


Figure II.15. Diagramme d'états-transitions des projets et de leurs résultats de gestion chez un producteur dans un cycle de gestion

A la réception d'une invocation du service, ce dernier vérifie le paramètre de confirmation *customer_confirmation* pour toutes les activités contenues dans la requête d'invocation. En fait, la valeur « Oui » de ce paramètre indique la confirmation du client pour le résultat proposé pour cette activité dans la réponse d'invocation précédente. Du fait que le client peut valider certaines activités d'un résultat pour un projet (confirmation partielle), il se peut qu'il y ait, dans une même requête d'invocation, des valeurs différentes du paramètre de confirmation pour des activités d'un même projet.

S'il n'y a pas d'activités confirmées dans la requête d'invocation, cela signifie que c'est une première invocation du service ou c'est un refus d'un résultat précédent, le projet enregistré dans la BDP a l'état *non traité*. Après le lancement de l'APE et l'établissement des résultats correspondants, les projets passent à l'état *traité* et les résultats établis ont l'état *non valide*.

S'il existe des valeurs « Oui » du paramètre *customer_confirmation* pour certaines activités d'un projet dans la requête d'invocation du service (confirmation partielle avant fin du cycle),

ce dernier vérifie la présence du résultat correspondant dans la BDP. Si le résultat est trouvé et son état est *non validé*, cela signifie que le projet existe dans la BDP et a l'état *traité*. Dans ce cas, les activités confirmées de ce résultat sont maintenues dans la BDP, l'état du projet et du résultat correspondants devient *validé*, et les ressources concernées sont réservées pour les prochains lancements de l'APE, et ce pour tout le long de la durée de validité du résultat. Pour les activités ayant comme valeur « Non » pour le paramètre *customer_confirmation*, le résultat est supprimé de la BDP, et l'état du projet devient *non traité* en saisissant les nouvelles contraintes pour les tâches demandant ces activités dans ce projet. Cela signifie que ces activités doivent être traitées dans le prochain lancement de l'APE. Dans la réponse d'invocation qui va être envoyée au client, le paramètre de confirmation du producteur (*producer_confirmation*) est initialisé à « Oui » pour les activités validées, et à « Non » pour les activités non validées pour lesquelles l'APE a établi un nouveau résultat.

Si toutes les activités contenues dans la requête d'invocation sont validées par le client (confirmation totale), ce qui correspond à une valeur « Oui » pour le paramètre *customer_confirmation* pour ces activités, et si les résultats correspondants existent encore dans la BDP et ont un état *non validé*, ces résultats sont maintenus dans la BDP ainsi que les projets correspondants. L'état de ces résultats et ces projets devient *validé*, et les ressources concernées sont réservées pour la durée de ces résultats. Ensuite, le service renvoie la réponse d'invocation contenant les résultats confirmés au client en indiquant la confirmation de cette validation par le producteur. Cette confirmation correspond au renvoi du même résultat de l'invocation précédente en remplaçant par « Oui » le paramètre de confirmation du producteur (*producer_confirmation*) pour toutes ces activités. Notons que le renvoi de ce résultat est fait avant le lancement de l'APE.

Dans le cas où la requête d'invocation comprend des activités validées par le client et que le service n'a pas trouvé les résultats concernés par cette validation dans la BDP (car résultats supprimés), le service considère que c'est une première invocation (le paramètre *customer_confirmation* est égal à « Non » pour toutes les activités dans la requête d'invocation). Cela correspond à la réception de la confirmation après la fin du cycle de gestion en cours. Pour cela, les projets correspondants sont stockés dans la BDP avec l'état *non traité* afin d'être traités dans le prochain lancement de l'APE. Même fonctionnement pour un refus total du résultat de l'invocation précédente qui correspond à des valeurs « Non » pour le paramètre *customer_confirmation* pour toutes les activités du projet présentes dans la

requête d'invocation. Dans ce cas, si les résultats existent encore dans la BDP, ils sont supprimés.

Normalement, la durée du stockage d'un résultat dans la BDP est de l'ordre d'un cycle de gestion. En effet, pour un résultat établi par le producteur, ce dernier ne peut pas attendre longtemps la réponse d'un client à son résultat car dans ce cas, il doit garder les ressources de production réservées pour plusieurs cycles tant que le client n'a pas encore répondu. Cette stratégie pénalise le producteur qui peut avoir des ressources saturées pour une longue durée et pour des résultats qui peuvent ne pas convenir au client. Cela n'étant pas dans l'intérêt du producteur du fait qu'après cette attente, le client peut refuser les résultats proposés.

A l'expiration du timeout pour le lancement de l'APE, l'administrateur de cette dernière récupère de la BDP les différents projets ayant comme état *non traité* et les charge dans l'APE, en supprimant tous les projets ayant l'état *traité* et les résultats ayant l'état *non validé*. Après le traitement, l'administrateur de l'APE enregistre dans la BDP les résultats de ce cycle de gestion avec l'état *non validé* et change l'état en *traité* pour les projets traités. Ensuite, le service récupère de la BDP ces résultats, génère une réponse contenant des valeurs « Non » pour le paramètre *producer_confirmation* pour les activités de ces résultats, et l'envoie à l'agent ambassadeur correspondant.

Les interactions avec un client se terminent quand il n'existe plus de projets (par conséquent des résultats) pour ce client dans la BDP du producteur, ou bien si tous les projets et les résultats pour ce client existant dans la BDP ont l'état *validé*.

4.4.5 Traitement et validation des résultats par le client

Chaque agent ambassadeur reçoit du producteur correspondant les résultats des services invoqués. Dans un projet client, plusieurs agents ambassadeurs peuvent être impliqués. Pour chaque cycle de gestion chez le client, il faut qu'il y ait un résultat de gestion pour tous les projets définis. Pour cela, nous considérons qu'à la fin de chaque cycle de gestion, il existe au moins un résultat de gestion pour chaque projet.

Le diagramme de séquences concernant le traitement et la validation d'un résultat chez le client est illustré sur la figure II.16.

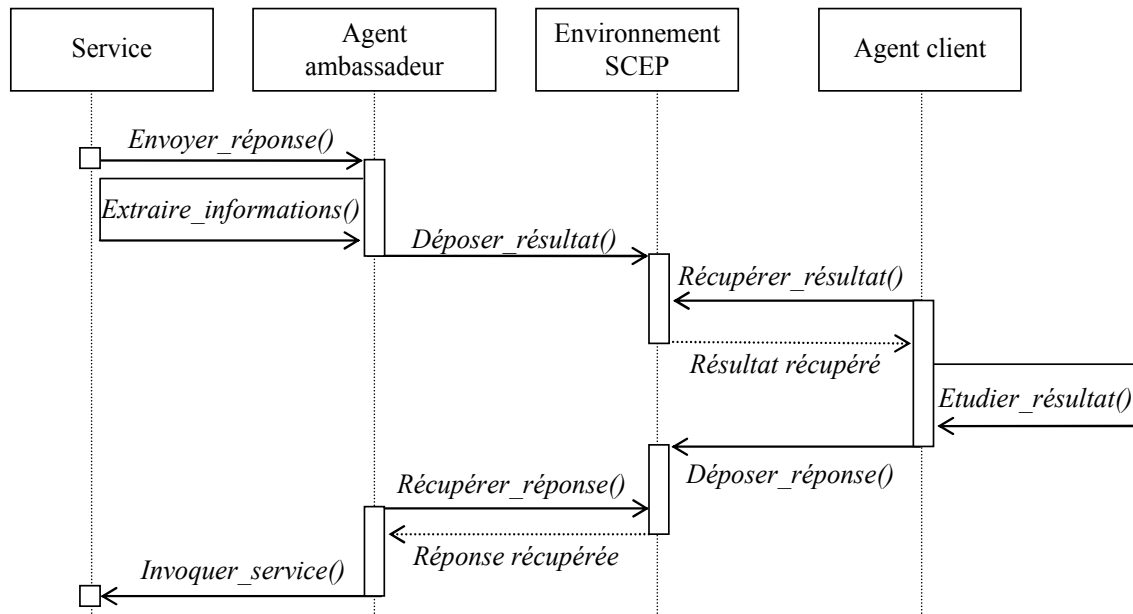


Figure II.16. Traitement et validation d'un résultat chez le client

La réponse contenant le résultat d'invocation du service reçu par un agent ambassadeur contient les identifiants des projets concernés, et les contraintes proposées par le producteur pour les activités demandées. Les informations contenues dans la réponse sont extraites par l'agent ambassadeur qui les dépose dans l'environnement SCEP pour être découvertes par les agents clients responsables des projets.

Dans la phase de validation, quatre cas peuvent avoir lieu : 1- le client confirme (valide) le résultat reçu de la part du producteur pour la totalité des activités, 2- il refuse le résultat complètement en choisissant celui d'un autre producteur, 3- il le refuse en proposant des nouvelles contraintes de réalisation souhaitées, ou 4- il annule des activités du projet ou confirme certaines activités du résultat et pas la totalité. Une ré-invocation du service est nécessaire dans tous ces cas sauf pour le deuxième.

Dans le premier cas qui concerne la validation du résultat reçu, le client doit renvoyer la réponse de confirmation au producteur avant l'expiration du timeout et la suppression du résultat de la BDP. Cela correspond à une ré-invocation du service pour le résultat confirmé en initialisant à « Oui » le paramètre de confirmation *customer_confirmation* pour les activités validées.

Dans le deuxième cas où le client refuse le résultat en choisissant celui fourni par un autre producteur, le client n'informe pas le producteur concerné par le refus. Du fait que le résultat

n'a pas été confirmé par le client chez le producteur correspondant, cela va conduire à la suppression de ce résultat de la BDP au prochain lancement de l'APE.

Dans le troisième cas, de nouvelles contraintes souhaitées sont définies pour les activités du service en initialisant à « Non » le paramètre de confirmation *customer_confirmation* pour ces activités. Cela permet au producteur d'interpréter cette invocation comme étant une première invocation et pas une confirmation d'une invocation précédente. Dans ce cas, les informations sont stockées dans la BDP du producteur et le résultat de l'invocation est établi dans le prochain cycle de gestion.

Dans le dernier cas qui est l'annulation d'une ou de plusieurs activités du projet, ou l'acceptation de certaines activités du résultat, si toutes les activités du résultat sont concernées par l'annulation, le producteur correspondant ne sera pas informé. Cela conduira à l'expiration de la durée de vie du résultat et à sa suppression de la BDP. Dans le cas où les activités annulées font partie d'un résultat convenable pour le client, ou en d'autres termes, si le client veut confirmer certaines activités d'un résultat et pas la totalité, il ré-invoque le service concerné en saisissant de nouvelles contraintes pour les activités non validées et en initialisant à « Non » le paramètre de confirmation *customer_confirmation* pour ces activités et à « Oui » celui des activités validées.

Les interactions avec un producteur se terminent dans deux cas. Le premier cas quand le client annule tous les projets demandant des services fournis par ce producteur. Le deuxième cas quand il a eu une réponse d'invocation de ce producteur confirmant toutes les activités validées. Cela correspond à une réponse contenant la valeur « Oui » du paramètre *producer_confirmation* pour toutes les activités contenues dans la requête d'invocation du service. Après la validation d'un résultat pour un producteur, le manager des projets clients enregistre les informations concernant ce producteur et les services qu'il fournit dans le registre local (RL) et incrémente le coefficient de collaboration pour ce producteur.

5 Conclusion

L'architecture SCEP-SOA proposée offre une gestion distribuée des projets de production multi-site, en faisant coopérer différentes applications de production. Elle met en œuvre les différents concepts mis en place dans un marché orienté-service, offrant des mécanismes de mise en relation et des collaborations dynamiques entre les partenaires grâce aux registres de

services (locale et globale). La présence du registre local chez le client permet de mieux guider ce dernier dans le choix de ses partenaires, offrant des possibilités de collaboration à long terme en permettant de passer d'une alliance temporaire d'entreprises telle que l'entreprise virtuelle, à une coopération plus stable avec les producteurs.

Au-delà de la mise en relation des clients et des producteurs, cette architecture masque à ces différents acteurs la complexité des réseaux d'applications à mettre en place pour réaliser les projets. Elle permet une coopération entre les différentes applications de production en offrant un couplage lâche entre les modèles de gestion manipulés par ces applications, ainsi qu'un fonctionnement indépendant de la plate-forme d'implémentation et une possibilité de réutilisation des services. Cette architecture doit être considérée comme une approche fédérée d'interopérabilité (Chen *et al.*, 2006) du fait qu'aucun partenaire n'impose rien aux autres partenaires. Chacun garde son propre mode de fonctionnement, ses technologies, ses logiciels et applications de production, et rend publique certaines fonctionnalités de ses applications en conservant les aspects privés du fonctionnement local.

Cependant, dans son état actuel, l'architecture SCEP-SOA ne prend pas en compte l'hétérogénéité qui peut exister au niveau des données et notamment dans la sémantique des informations et des concepts de production échangés entre les partenaires. L'intégration des concepts du modèle SOA seul dans l'architecture proposée ne garantit pas la bonne compréhension et la bonne interprétation des informations échangées entre les différentes applications de production utilisées par les partenaires. En effet, un concept de production peut avoir des significations qui peuvent différer d'un partenaire à l'autre. Pour appréhender cette hétérogénéité sémantique, de nouveaux composants doivent être rajoutés à cette architecture pour assurer l'interopérabilité sémantique, améliorant ainsi la communication entre les partenaires. Le chapitre suivant fera l'étude de l'interopérabilité sémantique dans l'architecture SCEP-SOA et des outils permettant son intégration à cette architecture.

Chapitre III

Interopérabilité sémantique dans SCEP-SOA

1 Introduction

L'architecture SCEP-SOA offre une gestion distribuée des projets multi-sites en faisant coopérer différentes applications et systèmes de production, indépendamment de l'implémentation et de la technologie utilisée par leurs applications. Cependant, rien ne garantit que les concepts utilisés par un partenaire pour modéliser les données de son application soient compréhensibles et bien interprétés par les autres partenaires. Pour appréhender cette problématique, la notion de sémantique des données échangées est abordée dans la section 2.

Les ontologies sont des outils pour représenter la sémantique d'un domaine de connaissance. Elles permettent de modéliser et spécifier, d'une manière formelle interprétable par les machines, les connaissances d'un domaine d'application. Ces ontologies, leurs définitions, leurs caractéristiques, leurs classifications ainsi que leurs intérêts et leurs apports dans différents domaines sont discutés dans la section 3.

Chaque entreprise a son propre vocabulaire et ses propres concepts de modélisation qui diffèrent au moins pour partie avec ceux des autres entreprises. Cela conduit à un problème d'hétérogénéité entre les différentes ontologies utilisées par les partenaires. Cette problématique d'hétérogénéité est discutée dans la section 4. La section 5 fait le point sur ces différentes techniques proposées pour gérer les hétérogénéités entre les ontologies.

Trois approches ont été proposées pour l'intégration des ontologies dans les architectures distribuées, à savoir l'approche mono-ontologique, l'approche multi-ontologique et l'approche hybride. Ces approches sont détaillées dans la section 6. Dans le contexte de l'architecture SCEP-SOA proposée, les approches multi-ontologique et hybride peuvent être appliquées. La section 7 décrit les deux stratégies proposées basées sur ces approches d'intégration pour assurer l'interopérabilité sémantique dans SCEP-SOA. Les techniques ontologiques utilisées, les nouveaux composants sémantiques ajoutés ainsi que le nouveau

fonctionnement de SCEP-SOA dans chacune de ces stratégies sont aussi discutés. Une comparaison des stratégies étudiées et une discussion sur la stratégie adoptée dans SCEP-SOA concluent ce chapitre.

2 La sémantique dans les problèmes d'interopérabilité

Dans le domaine des systèmes d'information, la sémantique réfère plus précisément au sens des différents éléments d'un système d'information : des données, des fonctions, voire des processus. La sémantique des éléments d'un système d'information peut différer d'un système à un autre, en fonction du contexte d'utilisation. Elle constitue un aspect fondamental de l'interopérabilité des applications d'entreprise. Il est nécessaire que les différentes applications utilisées puissent interpréter de la même manière les données échangées et/ou les fonctions réutilisées. Vernadat (Vernadat, 2006) considère l'interopérabilité sémantique comme la possibilité de partager, agréger ou synchroniser l'information dans des systèmes d'information hétérogènes. C'est donc une capacité à partager les informations d'un système d'information (processus, données et application) avec un autre système d'information. Les entreprises doivent donc bien décrire et définir leurs systèmes d'information pour partager efficacement leurs informations avec d'autres entreprises. Les conflits sémantiques d'après (Izza, 2006) ne concernent pas seulement les données, mais aussi les autres couches de l'intégration dont notamment les fonctions et les processus.

Selon Touzi, (2007) l'interopérabilité sémantique entre les systèmes d'information concerne les niveaux processus, données et applications. L'interopérabilité sémantique des processus concerne la possibilité de faire interopérer des processus appartenant à des domaines métiers différents, qui possèdent une structuration différente (contrôle de flux, structures itératives, etc.) ou modélisés avec des formalismes de processus divers. L'interopérabilité sémantique des données concerne la mise en commun des données hétérogènes de systèmes d'information, qui sont décrites dans des référentiels différents. L'interopérabilité sémantique des applications concerne la mise en commun des fonctions proposées par les applications des entreprises. Il peut exister, de la même manière que pour les données, une difficulté pour l'interprétation de la fonction exacte qui peut être rendue par une application.

En effet, les entreprises doivent d'abord bien définir, décrire et formaliser la connaissance liée à leurs systèmes d'information, afin de pouvoir la partager avec d'autres. La représentation de la sémantique est nécessaire pour l'intégration des systèmes. Pour cela, on utilise des

techniques de représentation de la connaissance. La représentation des connaissances étudie comment transformer l'expression du sens en une représentation formelle manipulable par une machine. Il existe principalement deux approches pour la modélisation de la sémantique : la modélisation procédurale et la modélisation déclarative. La modélisation procédurale se fait à travers l'utilisation de procédures ou de règles, tandis que l'approche déclarative se base sur la modélisation des faits. Même si l'approche procédurale de modélisation de la sémantique est une approche qui présente des performances très intéressantes en matière d'exécution, il n'en demeure pas moins que l'approche déclarative est généralement plus avantageuse en termes de standardisation, de capture, de réutilisation et d'inférence (Singh et Huhns, 2005). De plus, l'approche déclarative est plus flexible dans le sens où la modélisation ne dépend pas de l'ordre de représentation des faits, ce qui implique une modification plus aisée. L'un des outils les plus utilisés dans le cadre de la représentation déclarative de la sémantique et la conceptualisation de la connaissance de système d'information est la notion d'ontologie.

3 Ontologie et représentation de la sémantique

3.1 Définitions et langages pour l'ontologie

Plusieurs définitions ont été données pour l'ontologie. La première était celle donnée par Neches *et al.*, (1991) qui considèrent qu'une ontologie définit les termes basiques et les relations comportant le vocabulaire d'un secteur de matière aussi bien que les règles pour combiner des termes et des relations pour définir des extensions pour le vocabulaire. La définition de Gruber, (1993) est plus référencée dans la littérature et s'énonce comme suit: une ontologie est une spécification explicite d'une conceptualisation. Cette définition a été légèrement modifiée par Borst, (1997) en considérant qu'une ontologie est définie comme une spécification formelle d'une conceptualisation partagée.

Ces deux définitions ont été expliquées par (Studer *et al.*, 1998) : *conceptualisation* réfère à un modèle abstrait de certains phénomènes dans le monde en ayant identifié leurs concepts appropriés; *Explicite* signifie que le type de concepts utilisé et les contraintes de leur utilisation sont explicitement définis ; *Formel* se rapporte au fait que l'ontologie doit être compréhensible par une machine ; *Partagé* signifie qu'une ontologie capture la connaissance consensuelle, elle n'est pas privée pour certains individus mais plutôt acceptée par un groupe. Ces différents concepts ont été repris dans la définition donnée dans (Zhou et Kuntz, 2004) qui considère qu'une ontologie est une spécification explicite et partagée qui définit les concepts

primitifs, les relations, les fonctions, les axiomes et les instances comprenant une connaissance du sujet pour étendre, structurer et capter la connaissance explicite de ce sujet entre les gens, les organisations, les ordinateurs et les systèmes software.

Maedche, (2002) définit une structure pour l'ontologie, comprenant 5 concepts clés qui sont : les concepts, les relations, une fonction qui relie les concepts non taxonomiques, et un ensemble d'axiomes. Ces concepts sont définis comme suit :

$$O := \{C, R, H, ref, A^0\}$$

- Deux ensembles disjoints, C (concepts) et R (relations)
- Une hiérarchie de concepts, H : H est une relation directe $H \subseteq C \times C$ qui est appelée hiérarchie de concepts ou taxonomie. $H(C, C')$ signifie que C est un sous concept de C' .
- Une fonction $ref: R \rightarrow C \times C$ qui relie les concepts non taxonomiques
- Un ensemble d'axiomes A^0 , exprimé dans un langage logique approprié

Dans (Gruber, 1993) et (Gomez-Perez, 2000), les auteurs détaillent ces constituants de base d'une ontologie. Les *concepts*, aussi appelés termes ou classes de l'ontologie, constituent les objets de base manipulés par les ontologies. Ils correspondent aux abstractions pertinentes du domaine du problème, retenues en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie. Les *relations* traduisent les interactions existant entre les concepts présents dans le domaine analysé. Elles incluent la relation de spécialisation (subsomption), la relation de composition (méronymie), la relation d'instanciation, etc. Ces relations nous permettent de capturer la structuration ainsi que l'interaction entre les concepts, ce qui permet de représenter une grande partie de la sémantique de l'ontologie. Les *fonctions* sont des cas particuliers de relations dans lesquelles le $n^{ième}$ élément (extrant) de la relation est défini de manière unique à partir des $n-1$ éléments précédents (intrants). Les *axiomes* permettent de modéliser des assertions toujours vraies, à propos des abstractions du domaine traduites par l'ontologie. Ils permettent de combiner des concepts, des relations et des fonctions pour définir des règles d'inférences et qui peuvent intervenir, par exemple, dans la déduction, la définition des concepts et des relations, ou alors pour restreindre les valeurs des propriétés ou les arguments d'une relation. Les *instances* ou individus constituent la définition extensionnelle de l'ontologie. Ils représentent des éléments singuliers véhiculant les connaissances à propos du domaine du problème.

Bien que les définitions données pour l'ontologie soient différentes, elles considèrent néanmoins qu'une ontologie comprend les parties suivantes (Obitko et Marik, 2002):

- conceptualisation d'un domaine : une manière de visualiser ou modéliser un domaine
- spécification de cette conceptualisation : une description formelle

Au niveau de conceptualisation, nous décidons quels objets et relations parmi eux seront incluses dans l'ontologie et également à quel niveau de détails. Au niveau de spécifications, nous indiquons formellement la conceptualisation, habituellement en un certain langage formel. Les ontologies doivent être bien désignée et bien définies (Obitko et Marik, 2002). Bien désignée signifie qu'elle doit capturer d'une manière adéquate le domaine modélisé, être compréhensible pour un utilisateur humain et fournir un bon support pour le traitement d'une machine.

Les langages de définition d'une ontologie sont traités dans (Mota *et al.*, 2002). L'analyse qui a été faite montre que tous les cadres de représentation d'ontologie analysés manquent une manière normale et complète de déclarer des actions. Les fonctions et les attributs généraux peuvent être déclarés et définis seulement dans Ontolingua. Ontolingua est également le seul cadre permettant l'expression des axiomes arbitraires qui capturent des relations générales entre les entités de l'ontologie. Bien qu'Ontolingua soit essentiellement plus expressif que les autres approches analysées, il a des inconvénients : par exemple, il n'est pas possible de définir un procédé général d'inférence pour la puissance entière de la langue. Dans d'autres langages comme DAML (Darpa Agent Markup Language) et RDF (Resource Definition Framework), les concepts en tant que variables, fonctions ou paramètres d'action n'existent pas. Il n'est pas aussi facile de prolonger Ontolingua comme DAML et RDF. Actuellement, le langage d'ontologie de Web (Web Ontology Language ou OWL) est le plus extensivement utilisé pour éditer et partager des ontologies sur le Web afin de faciliter la gestion de la connaissance.

3.2 Classification des ontologies

Dans la littérature, il existe une classification des ontologies selon deux niveaux: le niveau de granularité (généralité) et le niveau de formalité (formalisme de représentation) (Uschold et Gruninger, 1996 ; Guarino, 1998).

Selon leur niveau de granularité, les ontologies peuvent être classifiées principalement en quatre catégories (Uschold et Gruninger, 1996 ; Guarino, 1997), comme illustré sur la figure III.1.

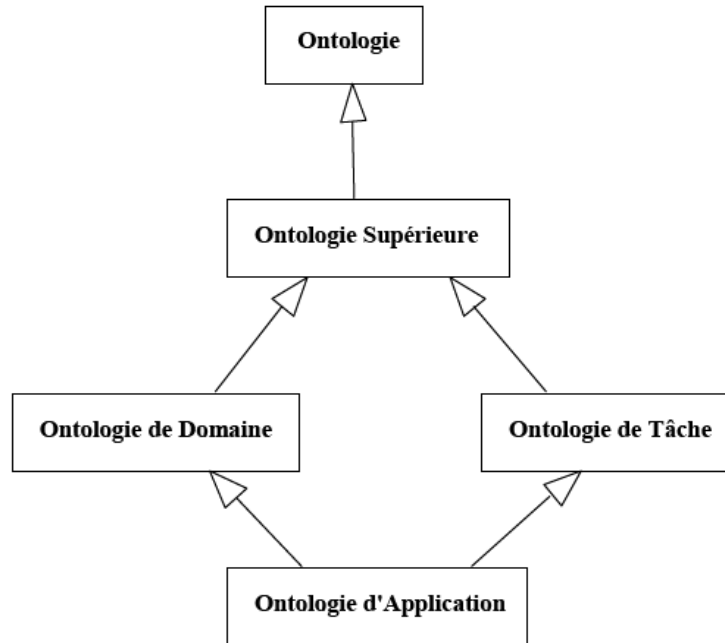


Figure III.1. Classification des ontologies selon le niveau de granularité

- Les ontologies supérieures (de haut niveau) (Guarino, 1999) ont pour objet l'étude des catégories de choses qui existent dans le monde, comme les concepts de haute abstraction tels que: les entités, les événements, les états, les processus, les actions, etc. et qui sont indépendants d'un domaine particulier.
- Les ontologies de domaine (Mizoguchi, 1995 ; Van Heijst *et al.*, 1997) sont des ontologies qui sont construites sur un domaine particulier de la connaissance. Elles fournissent le vocabulaire des concepts du domaine de connaissance et les relations entre ces derniers, les activités de ce domaine ainsi que les théories et les principes de base de ce domaine.
- Les ontologies de tâches (Mizoguchi, 1995) sont utilisées pour gérer des tâches spécifiques liées à la résolution de problèmes dans les systèmes, telles que les tâches de diagnostic, de planification, de configuration, etc. Elles fournissent un ensemble de termes au moyen desquels on peut décrire au niveau générique comment résoudre un type de problème.

- Les ontologies d'application (Van Heijst *et al.*, 1997) sont les ontologies les plus spécifiques. Elles permettent de décrire des concepts dépendant à la fois d'un domaine et d'une tâche. Dans cette classification, la notion d'ontologie d'application définit le contexte d'une application qui décrit la sémantique des informations et des services manipulés par un ensemble d'applications sur un même domaine.

Une autre classification des ontologies a été donnée par Van Heijst *et al.* (1997) qui ont classifié ces ontologies selon les critères de la quantité et le type de structure de la conceptualisation, et le sujet de la conceptualisation. Les premiers critères (quantité et structure) couvrent les ontologies de lexiques, d'informations, et de modélisation de la connaissance. Le deuxième couvre les ontologies de représentation, ontologies génériques, ontologies de domaines, et ontologies d'applications.

Concernant le niveau formel, Zhou et Kuntz, (2004) ont classifié d'abord la connaissance en : science naturelle et science sociale, puis la science naturelle en : connaissance basique, connaissance spécialisée, et connaissance d'application. Ils considèrent qu'une ontologie est illustrée comme un point dans un plan de coordonnées qui est représenté sur la figure III.2.

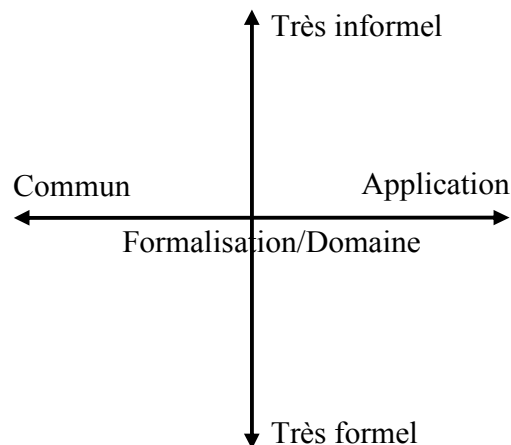


Figure III.2. Les ontologies dans un plan, adapté de (Zhou et Kuntz, 2004)

L'axe des sujets se rapporte à la généralité de la connaissance. L'axe de la formalité se rapporte à la formalité de la connaissance (Uschold et Gruninger, 1996). L'origine nous indique que l'ontologie est de domaine et de formalisation.

3.3 Apports des ontologies dans la gestion interopérable de la connaissance

Le but important de la construction d'une ontologie est de rendre la communication des connaissances plus facile (Malucelli, 2006). Les ontologies se sont développées et utilisées du fait qu'elles permettent entre elles de partager la connaissance en partageant l'arrangement de la structure d'information partagée entre des agents logiciels et des personnes. Elles permettent également la réutilisation de la connaissance en permettant la réutilisation de l'ontologie pour des systèmes opérant dans un domaine similaire. En outre, elles permettent de rendre explicites les principes et les hypothèses sur un domaine, pour une communication plus facile. Wongthongtham et Chang, (2004) considèrent que le but d'une ontologie est la publication et la communication de la connaissance. Les ontologies ont été employées pour différents objectifs et dans divers domaines, *i.e.* la communication, l'ingénierie des systèmes, le Web sémantique, les systèmes multi-agents (SMA), etc.

La communication peut avoir lieu entre les hommes et/ou les systèmes. Les ontologies permettent alors le partage de la compréhension et la communication dans des contextes particuliers et selon les besoins. Ainsi, on peut utiliser l'ontologie pour créer un réseau de relations qui définit les connexions entre les composants du système. Cette caractéristique de communication est offerte grâce à la non-ambiguïté des termes utilisés et définis par l'ontologie dans les systèmes.

Dans l'ingénierie des systèmes, les ontologies jouent un rôle important sur trois aspects : la spécification, la fiabilité et la réutilisation. Une ontologie peut aider à l'analyse des besoins et à définir les spécifications d'un système. Son rôle dépend du degré de la formalisation et l'automatisation de la méthode de spécification. Dans l'aspect informel, elles facilitent l'identification des besoins du système logiciel. Elles facilitent également la compréhension des liens et des relations entre les composants de ce système. Pour l'aspect formel, elles définissent la spécification déclarative du système qui fournit des caractéristiques de l'ontologie indépendamment de la façon dans laquelle elle est implémentée. Les ontologies informelles améliorent et assurent la fiabilité des systèmes logiciels en servant de base pour la vérification manuelle de la conception. Elles permettent la vérification semi-automatique du système en respectant la spécification déclarative et l'intégration des différents composants du système.

Les ontologies doivent être réutilisables (Mellal, 2007). Néanmoins, un problème est rencontré quand les systèmes logiciels sont appliqués dans de nouveaux domaines : les ontologies satisfont les applications d'origine, mais non les nouvelles. Parmi les solutions proposées dans (Charlet *et al.*, 2000), on trouve la construction d'une librairie d'ontologies réutilisables et adaptables dans différents environnements.

Dans les SMA, Obitko et Marik, (2002) comparent le rôle des ontologies dans le domaine de la fabrication avec celles proposées pour le Web sémantique. Une des différences principales est que les ontologies dans le domaine de la fabrication tendent vers la standardisation et la fixation dans le but d'achever un comportement stable et garanti. Breitman *et al.*, (2005) utilisent les ontologies pour formaliser et spécifier les services dans les SMA. Leurs travaux se concentrent sur OWL (Web Ontology Language) pour permettre l'interopérabilité entre des applications software, des services Web, et des agents intelligents.

Le "E-commerce" est une des applications les plus répandues des ontologies. En général, elles sont utilisées pour déterminer les index conceptuels qui décrivent les ressources sur le Web. Le Web sémantique se base sur les ontologies pour formaliser les concepts du domaine qui sont partagés par les services (Srivastava, 2003).

4 Problématique de l'interopérabilité sémantique dans SCEP-SOA

L'interopérabilité sémantique consiste à donner un sens aux informations échangées et à garantir que ce sens est distribué dans tous les systèmes entre lesquels des échanges doivent être mis en œuvre (Vernadat, 2006). Elle pose, de ce fait, un problème de compréhension des informations échangées entre des systèmes coopérant à la réalisation d'une tâche globale. Les ontologies sont un moyen contribuant à faciliter la compréhension des informations échangées entre les systèmes en essayant de standardiser la représentation des concepts et de leurs relations.

Cependant, dans un environnement de collaboration distribuée qui est le cas de la production multi-site, chaque partenaire a une ontologie pour son système de production, qui peut être différente des autres ontologies manipulées par les autres systèmes de production. La production multi-site nécessite une communication et une collaboration entre les différents partenaires, où chaque partenaire a sa propre vision du domaine et doit interagir avec d'autres partenaires en partageant ses connaissances. Pour mener à bien la collaboration entre les

partenaires, il faut s'assurer que les informations échangées entre ces partenaires soient compréhensibles et interprétées de la bonne manière. En d'autres termes, il faut résoudre l'hétérogénéité qui peut exister entre les différentes ontologies manipulées par les différentes applications des partenaires.

4.1 Classification de l'hétérogénéité entre les ontologies

Pour assurer l'interopérabilité entre des ontologies hétérogènes, Tang *et al.*, (2005) considèrent qu'il faut résoudre les deux problèmes suivants : l'hétérogénéité des méta-données et l'hétérogénéité des instances. L'hétérogénéité des méta-données concerne le sens des informations décrites par l'ontologie. Dans ce problème d'hétérogénéité, deux types de conflit peuvent être identifiés : le conflit de structure qui représente la différence entre les structures sémantiques des concepts manipulés par les ontologies dans un même domaine de connaissance, le conflit de noms qui représente l'utilisation d'un même nom par plusieurs concepts différents ou l'utilisation de plusieurs noms différents pour désigner un même concept. L'hétérogénéité des instances ou hétérogénéité des représentations concerne les différentes manières de représenter les informations décrites dans une ontologie. Par exemple le nom d'une personne peut être représenté par "Karim Ishak" ou par "Ishak Karim".

Selon De Bruijn *et al.* (2006), les deux types basiques d'hétérogénéité entre les ontologies sont : la différence de conceptualisation due aux différents concepts utilisés pour modéliser le même domaine, et la différence d'explication qui représente les différentes manières de spécifier cette conceptualisation. La différence de conceptualisation contient deux catégories : une différence de portée qui a lieu quand deux classes différentes ont des chevauchements dans leurs extensions (ensembles des instances), et la différence entre la couverture du modèle et la granularité qui est représentée par une différence dans la partie du domaine couvert par deux ontologies. Par exemple, une ontologie peut avoir un concept *Personne* et une autre ontologie peut distinguer entre *JeunePersonne*, *PersonneDeMoyenAge*, et *VieillePersonne*.

Dans la différence d'explication, il y a trois catégories : la différence dans le style de la modélisation dans le cas où le paradigme utilisé pour spécifier certain concept, *i.e.* temps, est différent, *i.e.* intervalles vs points de temps, ou la manière de la description du concept diffère, *i.e.* utiliser des sous-classes vs des attributs pour distinguer les groupes d'instances. La deuxième catégorie est la différence terminologique où deux concepts équivalents sont représentés en utilisant des noms différents ou un même nom est utilisé pour différents

concepts. La troisième catégorie est la différence d'encodage ou la différence d'échelle qui a lieu quand les valeurs dans les différentes ontologies sont encodées d'une manière différente, *i.e.* mesure de la distance en kilomètres ou en miles.

Dans (Saleem, 2006), l'hétérogénéité sémantique entre les ontologies est divisée en trois catégories : conflit structurel qui a lieu quand les ontologies possèdent des structures sémantiques différentes pour définir un même concept, conflit de nom qui a lieu quand un même terme est représenté par des noms différents dans des ontologies différentes, conflit de représentation de données qui a lieu quand une même information ou donnée a des représentations différentes. Une autre classification de l'hétérogénéité sémantique est donnée dans (Pluempitiwiriyawej et Hammer, 2000). Cette classification comprend trois grandes classes : les conflits structurels (conflits de généralisation/spécification, conflits de noms, manque d'items, ordres des éléments, différence de types de données, etc.), les conflits de domaine (différences des schémas, différences d'échelles ou d'unités, différences dans la représentation des données, etc.), les conflits des données (orthographes incorrectes, manque de données, conflits de noms entre le contenu d'un élément et la valeur d'un attribut, etc.).

L'hétérogénéité sémantique est classée en quatre niveaux dans (Bouquet *et al.*, 2005). Ces niveaux sont : le niveau syntaxique, le niveau terminologique, le niveau conceptuel, et le niveau sémiotique ou pragmatique. Le niveau syntaxique représente la différence dans le langage de description de l'ontologie (OWL, KIF, etc.). Le niveau terminologique représente le conflit de nom (synonymie, polysémie, etc.) qui peut exister entre les éléments des ontologies. Le niveau conceptuel est représenté par deux classes de différences : les différences métaphysiques qui concernent la division du monde en pièces (quelles entités, propriétés, et relations sont représentées dans l'ontologie), et les différences épistémiques qui concernent les assertions qui sont faites sur les entités sélectionnées. Finalement, le niveau sémiotique ou pragmatique concerne le fait que différents individus et communautés peuvent interpréter la même ontologie d'une manière différente et dans des contextes différents.

L'étude faite dans (Nagarajan *et al.*, 2006) montre une classification de l'hétérogénéité en quatre niveaux :

- L'hétérogénéité syntaxique : représente la différence dans le langage utilisé pour représenter les éléments.

- L'hétérogénéité structurelle : représente la différence dans les types et les structures des éléments.
- L'hétérogénéité représentationnelle ou des modèles : représente la différence entre les modèles (base de données, ontologie) et leurs représentations (relationnelle, orientée-objet, RDF, OWL).
- L'hétérogénéité sémantique : qui représente la différence de représentation d'une même entité réelle où plusieurs termes ou structures sont données pour une même entité ou *vice versa*.

Compte tenu de ces différentes classifications, nous présentons sur la figure III.3 la classification pour l'hétérogénéité entre les ontologies faisant la synthèse des classifications présentées.

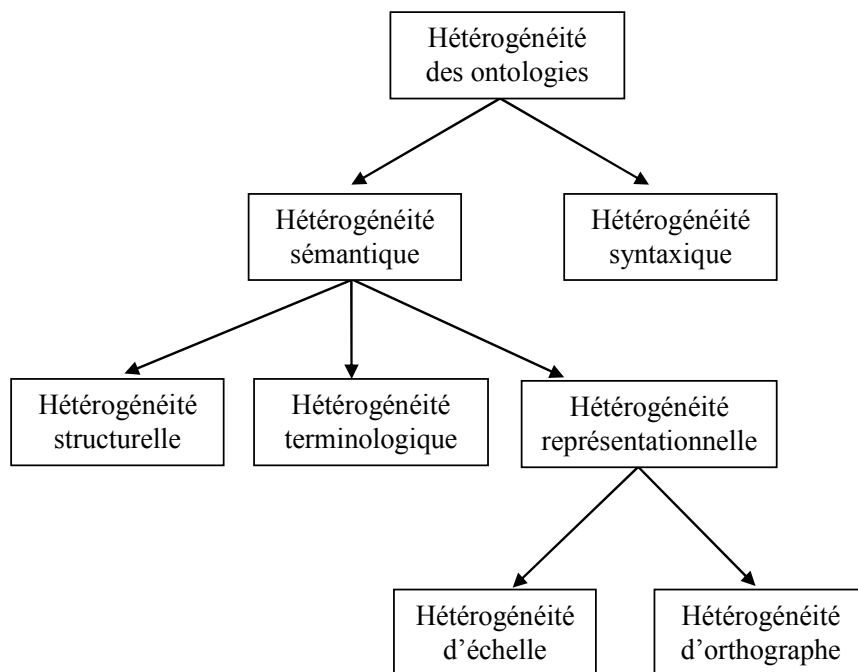


Figure III.3. Synthèse sur la classification de l'hétérogénéité des ontologies

Elle montre que l'hétérogénéité entre les ontologies comprend deux grandes classes qui sont : l'hétérogénéité syntaxique et l'hétérogénéité sémantique.

L'hétérogénéité syntaxique correspond à celle mentionnée dans (Bouquet *et al.*, 2005 ; Nagarajan *et al.*, 2006). Elle représente la différence entre les langages de définition utilisés pour représenter les ontologies de production (DAML, OWL, KIF, etc.).

L'hétérogénéité sémantique regroupe trois classes d'hétérogénéités: structurelle, terminologique et représentationnelle.

L'hétérogénéité structurelle représente la différence entre les structures des éléments. Elle correspond à celle mentionnée dans (Tang *et al.*, 2005 ; Saleem, 2006 ; Pluempitiwiriyaewej et Hammer, 2000 ; Nagarajan *et al.*, 2006).

L'hétérogénéité terminologique correspond au conflit de nom mentionné dans toutes les classifications présentées précédemment.

Les deux types d'hétérogénéité sémantique déjà présentés (structurelle et terminologique) sont plus liés à la définition et à la spécification de l'ontologie en termes de classes et propriétés (attributs). L'hétérogénéité représentationnelle concerne le contenu ou les instances de ces classes et propriétés et représente la différence dans la représentation des instances des différentes ontologies. Elle a été considérée dans toutes les classifications présentées sauf celle donnée dans (Bouquet *et al.*, 2005). Elle comprend l'hétérogénéité d'échelle qui concerne l'utilisation des différentes échelles par les instances et considérée notamment dans (Pluempitiwiriyaewej et Hammer, 2000 ; De Bruijn *et al.*, 2006), et l'hétérogénéité d'orthographe qui concerne la différence dans l'écriture d'une même instance.

4.2 Hétérogénéités des ontologies dans SCEP-SOA

Nous considérons la classification proposée sur la figure III.3. Dans SCEP-SOA, l'hétérogénéité syntaxique n'est pas prise en compte du fait que nous considérons que toutes les ontologies mises en place chez les différents partenaires sont décrites en OWL qui est le langage de définition le plus utilisé. Nos travaux traitent la problématique de l'hétérogénéité sémantique qui représente la différence de signification entre les concepts modélisés par chaque ontologie. Dans SCEP-SOA, cette hétérogénéité concerne les trois types d'hétérogénéité présentés précédemment: hétérogénéité structurelle, hétérogénéité terminologique, et hétérogénéité représentationnelle.

L'hétérogénéité structurelle est illustrée par la différence entre les structures des concepts manipulés par les ontologies où un même concept peut avoir des hiérarchies de structure différentes. Un exemple de cette hétérogénéité est donné sur la figure III.4 qui présente deux structurations différentes d'un même concept (Machine) dans deux ontologies différentes dans le domaine de la planification et l'ordonnancement de la production.

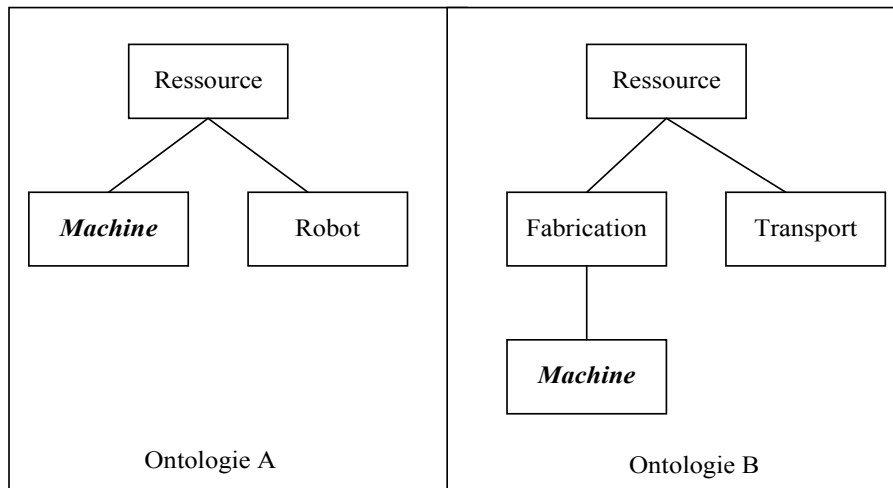


Figure III.4. Illustration de l'hétérogénéité structurelle dans SCEP-SOA

Aussi, cette hétérogénéité (structurelle) comprend le conflit attribut/entité où un concept est modélisé comme une classe dans une ontologie et comme un attribut d'une classe dans une autre *i.e.* concept *Localité* pour une ressource qui peut être représentée comme une classe à part ou comme une propriété de la classe *Ressource*.

L'hétérogénéité terminologique est illustrée par l'utilisation de noms ou des termes différents pour désigner un même concept, *i.e.* le concept activité de production peut être représenté par plusieurs termes tels que opération, tâche, etc. Chaque terme dépend de la terminologie utilisée par le partenaire dans le fonctionnement de son système de production.

L'hétérogénéité représentationnelle est illustrée par l'hétérogénéité d'échelle, *i.e.* coût d'une activité de fabrication exprimé en \$ ou en £ ou en €, etc., et l'hétérogénéité d'orthographe, *i.e.* date de début d'une activité exprimée en année/moi/jour ou en jour-moi-année, etc.

Du fait de l'existence de ces hétérogénéités sémantiques entre les ontologies de production utilisées par les différents acteurs de l'architecture SCEP-SOA, des mécanismes et techniques d'interopérabilité (médiations d'ontologies) sont nécessaires pour appréhender cette hétérogénéité et permettre une meilleure collaboration et communication entre les partenaires. Plusieurs techniques de médiation d'ontologies existent, nous les détaillons dans ce qui suit.

5 Techniques d'interopérabilité des ontologies

Bien que les ontologies représentent un outil de partage de la connaissance, néanmoins, elles ne permettent pas de garantir la bonne interprétation et la compréhension de ces connaissances du fait de leur hétérogénéité. Dans la section précédente, nous avons montré

que les ontologies ont beaucoup de points faibles qui les rendent hétérogènes à différents niveaux. Pour résoudre ces hétérogénéités, de nombreuses approches ou méthodologies ont été proposées pour appréhender cette problématique. La médiation d'ontologie est une branche de recherche qui concerne la détermination et la résolution des problèmes d'hétérogénéité entre les ontologies en vue de la réutilisation de ces ontologies, ainsi que les données annotées selon ces ontologies, à travers des applications hétérogènes (De Bruijn *et al.*, 2006). Ces médiations d'ontologies sont classées en trois catégories : la correspondance d'ontologies, l'alignement d'ontologies, et la fusion d'ontologies. Il s'ajoute à ces médiations la catégorie de traduction ou translation d'ontologies. Ces différentes techniques sont détaillées dans la suite.

5.1 Correspondance d'ontologies

Ehrig et Staab, (2004) définissent le mécanisme de correspondance entre deux ontologies O1 (source) et O2 (cible) comme étant la manière à trouver, pour chaque entité (concept, relation, ou instance) dans l'ontologie O1, une entité correspondante qui ait le même sens dans l'ontologie O2. La correspondance peut être unidirectionnelle c'est-à-dire de O1 vers O2 par exemple ou bidirectionnelle c'est-à-dire de O1 vers O2 et *vice versa*.

Le mécanisme de correspondance d'ontologies comprend trois étapes principales (Noy, 2004):

1. Découverte des correspondances : consiste à chercher les similarités entre deux ontologies. Déterminer quels sont les concepts et les propriétés qui représentent des notions similaires.
2. Représentations formelles et déclaratives des correspondances : concerne l'exploration des manières possibles pour représenter les correspondances en vue de raisonner avec elles.
3. Reasonner avec les correspondances : consiste à trouver le type de raisonnement et ce qu'on veut faire avec les correspondances une fois qu'elles sont définies.

La découverte des correspondances est considérée la tâche la plus difficile et deux architectures ont été identifiées pour découvrir ces correspondances. La première consiste en la construction d'une ontologie partagée et générale qui convient aux développeurs des différentes applications, ensuite ces développeurs étendent cette ontologie générale avec des

concepts et des propriétés spécifiques à leurs applications, ce qui facilite la découverte des correspondances entre deux extensions grâce à cette ontologie partagée. La deuxième architecture est basée sur les techniques heuristiques et l'apprentissage des machines (machine learning) qui utilisent des caractéristiques variées des ontologies, comme leurs structures, leurs définitions des concepts, leurs instances de classes pour trouver les correspondances.

Concernant la représentation ou l'expression des correspondances entre les ontologies, ces correspondances peuvent être représentées sous formes d'instances dans une ontologie de correspondances, ou bien définir des axiomes de pont (bridging axioms) dans la logique du premier ordre pour représenter les transformations, ou bien utiliser des vues pour décrire les correspondances d'une ontologie globale vers les ontologies locales. Une fois les correspondances entre les ontologies représentées, elles peuvent être utilisées dans beaucoup de tâches d'intégration comme les demandes des requêtes, la composition des services Web, et notamment pour la transformation des données qui est le cas de notre contexte où nous cherchons à traduire des données entre différentes ontologies de production. De nombreux outils logiciels permettent la découverte et la représentation des ontologies, une étude de ces outils est donnée dans (Saleem, 2006). Une évaluation de ces outils dans (Noy et Musen, 2002).

5.2 Alignement d'ontologies

Nous avons déjà introduit que l'alignement d'ontologies est une des catégories des médiations d'ontologies. Une définition de l'alignement d'ontologies est donnée dans (Abolhassani *et al.*, 2006) comme étant le processus de trouver les entités en relation entre les différentes ontologies. Une autre définition est donnée dans (De Bruijn *et al.*, 2006) qui permet de déduire une certaine relation entre l'alignement d'ontologies et le mapping d'ontologies. Cette définition considère l'alignement d'ontologie comme un processus de découverte de similarités entre deux ontologies sources. Cette définition nous rappelle l'étape de découverte des correspondances dans le mapping des ontologies ce qui nous permet de considérer, en partant de ce point de vue, que l'alignement d'ontologie concerne la première étape dans le mapping d'ontologies. De nombreuses algorithmes ont été proposés pour cet objectif, une classification de ces algorithmes est donnée dans (De Bruijn *et al.*, 2006).

L'alignement d'ontologies concerne aussi des ontologies faiblement structurées ou se limitant à des simples hiérarchies de classification. Ces techniques d'alignement sont basées sur la structure d'une ressource complémentaire qui est représentée le plus souvent par une ontologie de support dite *background ontology* pour compenser le manque de structure dans les ontologies à aligner (Aleksovski *et al.*, 2006 ; Sabou *et al.*, 2006 ; Kalfoglou et Hu, 2005 ; Reynaud et Safar, 2006). L'idée de base de cette approche est d'aligner d'abord les concepts des ontologies source et cible avec la connaissance complémentaire (*background ontology*), ce qui constitue la phase d'ancrage, ensuite utiliser la structure de cette connaissance complémentaire pour déduire les relations sémantiques entre les concepts sources et cibles (phase de dérivation), et finalement utiliser ces relations pour déduire les correspondances entre ces concepts. Une analyse comparative entre les travaux utilisant cette approche de connaissance complémentaire est donnée dans (Safar *et al.*, 2007). Cette étude montre que cette approche est intéressante lorsqu'on connaît précisément le contexte au sein duquel les éléments manipulés doivent être interprétés et qu'il est possible de le spécifier. WordNet est souvent utilisé comme ressource complémentaire car il est une bonne source d'information sur les synonymies et fournit une hiérarchie de concepts basée sur les relations de généralisation/spécialisation. Cependant, l'étude faite dans (Safar *et al.*, 2007) a montré qu'il est difficile d'en extraire des relations pertinentes si le sens précis des termes cherchés n'est pas préalablement défini.

5.3 Fusion d'ontologies

La fusion d'ontologies (*ontology merging*) est le processus de création d'une ontologie à partir de deux ou plusieurs ontologies sources (De Bruijn *et al.*, 2006). L'ontologie résultante unifie les concepts présents dans les ontologies sources. Le mécanisme de fusionnement d'ontologies est appliqué dans le cas où les deux ontologies modélisent des concepts dans le même domaine ou dans des domaines similaires. Deux approches sont identifiées pour la fusion d'ontologies (De Bruijn *et al.*, 2006). La première approche consiste à prendre une collection d'ontologies et produire une nouvelle ontologie fusionnée qui capture les ontologies d'origine. L'ontologie résultante remplace les ontologies sources. Un exemple de cette approche est le système PROMPT (Noy et Musen, 2000) qui est un algorithme et un outil pour fusionner deux ontologies.

Dans la deuxième approche, les ontologies sources ne sont pas remplacées, une ontologie dite de pont (*bridge ontology*) est créée, qui importe les ontologies sources et relie les concepts de

ces ontologies en utilisant un nombre d'axiomes de pont (bridge axioms). Un exemple de cette approche est le système OntoMerge (Dou *et al.*, 2002). Ce système est une approche en ligne dans laquelle les ontologies sources sont conservées à l'opposition du système PROMPT. Une autre différence avec la première approche est que dans OntoMerge, le résultat de l'opération de fusion n'est pas une ontologie complètement fusionnée comme dans PROMPT, mais plutôt une ontologie de ponts qui contient des règles de translation (bridging axioms) pour connecter les parties similaires dans les ontologies sources.

De nombreux travaux ont été faits dans le domaine de fusion d'ontologies. Stumme et Maedche, (2001) proposent une nouvelle méthode FCA-Merge basée sur l'approche ascendante qui offre une description structurelle et globale du processus de fusion d'ontologies. Cette méthode consiste à extraire les instances d'un ensemble de documents textes dans un domaine spécifique en appliquant les techniques de traitement du langage naturel.

Dans le contexte des ontologies de domaines similaires, Mohsenzadeh *et al.*, (2005) proposent une nouvelle approche de fusion d'ontologies qui est différente de la plupart des approches de fusion qui sont adressées aux experts du domaine des ontologies. Cette approche permet d'obtenir les similarités syntaxiques et sémantiques entre les ontologies en les comparant ensemble sans utiliser une ontologie globale.

En se focalisant sur la découverte des interprétations informelles des concepts spécifiés dans une ontologie, l'approche HCONE (Kotis et Vouros, 2004) permet de valider les correspondances établies entre les ontologies à fusionner et trouver l'ensemble minimal d'axiomes pour l'ontologie résultante. La connaissance linguistique et structurelle sur les ontologies est exploitée par la méthode LSI (Latent Semantics Indexing) (Deerwester *et al.*, 1990) pour associer les concepts à leurs interprétations humaines informelles obtenues de WordNet. En utilisant les sémantiques des concepts, la méthode proposée traduit les définitions formelles des concepts à un vocabulaire commun et exploite les définitions traduites par le biais des services de raisonnement basés sur les logiques de descriptions.

5.4 Traduction d'ontologies

Selon Corcho, (2002) le problème de traduction ou translation d'ontologie apparaît quand on décide de réutiliser une ontologie ou une partie d'une ontologie en utilisant un outil ou un langage qui est différent de ceux dans lesquels cette ontologie est disponible. Il classe les

problèmes de translation en quatre catégories : problèmes lexicaux, problèmes syntaxiques, problèmes d'expressivité, et problèmes de raisonnement. Trois types de problèmes de translation d'ontologies sont identifiés dans (Dou *et al.*, 2005) et qui sont : translation d'ensembles de données (datasets) d'une ontologie à une autre, génération des extensions d'une ontologie, et établir des requêtes à travers plusieurs ontologies où la réponse à une requête peut utiliser des connaissances qui sont décrites selon différentes ontologies différentes de celle utilisée par la formulation de la requête. Pour répondre à ces problèmes, les auteurs dans (Dou *et al.*, 2005) proposent le système en ligne OntoMerge qui permet une translation entre les ontologies en se basant sur la fusion d'ontologies et les axiomes de pont (bridging axioms) pour automatiser le raisonnement sur les correspondances entre les ontologies à fusionner.

Dans le contexte du web sémantique, une approche de service Web de translation d'ontologies exprimées en OWL a été présentée dans (Mota et Botelho, 2005). Cette approche est basée sur le cadre de représentation d'ontologies O_3F (Object Oriented Ontology Framework). Ce cadre permet de représenter les ontologies en utilisant un modèle orienté objet et introduit des définitions de plusieurs niveaux de *traductibilité* d'ontologies : fortement translatables, extension, équivalentes, identiques, faiblement translatables, approximativement translatables. Le processus de translation proposé comprend deux tâches. D'abord, la vérification de la *traductibilité* des concepts définis dans l'ontologie, ensuite, translation des expressions conformément à l'ontologie. Ce processus est implémenté à travers un service Web qui répond à deux types de requêtes : détermination des niveaux de translation, et translation des expressions RDF entre les ontologies.

6 Approches d'intégration des ontologies

Afin d'assurer l'interopérabilité sémantique entre différentes sources d'information, plusieurs manières d'intégration et d'utilisation des ontologies ont été proposées. Dans cet objectif, trois directions ou approches d'intégration d'ontologies peuvent être identifiées (Wache *et al.*, 2001) : l'approche mono-ontologie (single ontology approach), l'approche multi-ontologie (multiple ontologies approach), et l'approche hybride (hybrid approach). Ces trois approches sont illustrées sur la figure III.5.

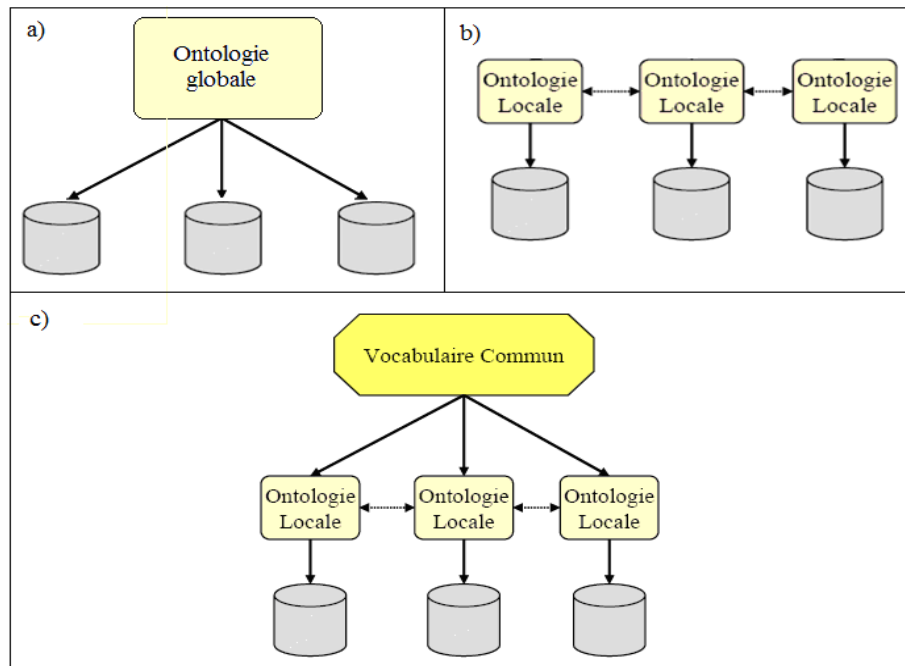


Figure III.5. Les trois approches pour l'intégration des ontologies (adapté de (Wache et al., 2001))

L'approche mono-ontologie représentée sur la figure III.5a utilise une ontologie globale qui peut être aussi une combinaison de plusieurs ontologies spécialisées, offrant un vocabulaire partagé pour la spécification de la sémantique utilisée par les sources d'information. Toutes les sources d'information sont issues de l'ontologie globale. L'approche mono-ontologie peut être appliquée dans les problèmes d'intégration dans le cas où toutes les sources d'information à intégrer ont presque la même vue du domaine. Si une des sources possède une vue différente, l'existence d'une ontologie commune devient une tâche très difficile (Gruber, 1995). Cependant, l'approche mono-ontologie est susceptible aux changements dans les sources d'information qui peuvent affecter la conceptualisation du domaine représenté dans l'ontologie. Selon la nature des changements d'une source d'information, il peut y avoir des changements de l'ontologie globale ainsi que des correspondances avec d'autres sources d'information. Ces inconvénients ont mené au développement de l'approche d'ontologies multiples ou multi-ontologie.

Dans l'approche multi-ontologie présentée sur la figure III.5b, chaque source d'information est décrite par sa propre ontologie. En fait, l'ontologie d'une source d'information peut être une combinaison de plusieurs autres ontologies. Les ontologies sont différentes et ne partagent pas le même vocabulaire comme dans l'approche mono-ontologie. Dans cette approche, chaque ontologie source est développée indépendamment de toute autre ontologie ou vocabulaire, aucune ontologie commune avec l'accord de toutes les sources n'est

nécessaire. En effet, l'absence d'un vocabulaire commun rend très difficile la comparaison des différentes ontologies. Dans cet objectif, l'approche d'ontologie dite hybride a été proposée.

L'approche hybride, présentée sur la figure III.5c, tire avantages des deux autres approches d'ontologies. Comme dans l'approche multi-ontologie, la sémantique de chaque source d'information est décrite selon sa propre ontologie. Afin de rendre les ontologies sources comparables, elles sont construites selon un vocabulaire global et commun, qui lui-même peut être aussi une ontologie. Ce vocabulaire partagé contient les concepts de base d'un domaine. Pour construire les termes complexes d'une ontologie source, les concepts primitifs sont combinés par des opérateurs. Du fait que chaque terme d'une ontologie source est basé sur des concepts primitifs du vocabulaire partagé, les termes deviennent comparables plus facilement que dans l'approche multi-ontologie. Cette approche est aussi caractérisée par la facilité de l'ajout de nouvelles ontologies sources sans avoir besoin de modifier le vocabulaire commun ou les correspondances ontologiques déjà établies.

7 Stratégies d'interopérabilité sémantique dans SCEP-SOA

Dans SCEP-SOA, il est considéré que chaque acteur (client ou producteur) possède déjà sa propre ontologie et qui n'est pas nécessairement la même utilisée par les autres acteurs. Cela a été illustré précédemment dans la problématique de l'interopérabilité sémantique dans SCEP-SOA montrant les hétérogénéités qui peuvent exister entre les différentes ontologies des partenaires. Du fait que toutes les sources d'information n'ont pas la même vue du domaine de la production et que les systèmes de production utilisés par les partenaires ne partagent pas le même vocabulaire, l'approche mono-ontologie ne peut pas être appliquée dans SCEP-SOA. Des stratégies basées sur l'application des deux autres approches (multi-ontologie et hybride) sont discutées dans la suite afin d'assurer l'interopérabilité sémantique dans cette architecture.

7.1 Stratégie basée sur l'approche multi-ontologie

L'idée de cette stratégie est d'établir des correspondances ontologiques chez les clients uniquement, permettant aux différents producteurs (normalement des PME) de garder leur propre ontologie de production sans se soucier des autres ontologies (celles des clients) et. L'application de l'approche multi-ontologie dans cette stratégie masque aux producteurs les mécanismes de correspondances ontologiques établies par les clients facilitant ainsi l'échange

d'informations entre les différents acteurs dans SCEP-SOA. Chaque client établit des correspondances bidirectionnelles entre sa propre ontologie de production et celles utilisées par les différents producteurs avec lesquels il collabore. Dans cette stratégie, les composants sémantiques ajoutés dans l'architecture SCEP-SOA se limitent au registre de services et aux clients. Le nouveau fonctionnement de cette architecture intégrant cette stratégie comprend deux étapes. La première consiste en la découverte et la représentation des correspondances entre les ontologies. Elle est réalisée lors de la publication des services par les producteurs et leur découverte par les clients. La deuxième étape concerne le raisonnement avec ces correspondances lors de l'interaction entre les clients et les producteurs pour la gestion de la réalisation des projets de production multi-sites.

L'établissement des correspondances ontologiques nécessite d'abord une connaissance des différentes ontologies mises en jeu. Afin de les identifier, un serveur d'ontologies (SO) est mis en place au niveau du registre des services comme illustré sur la figure III.6.

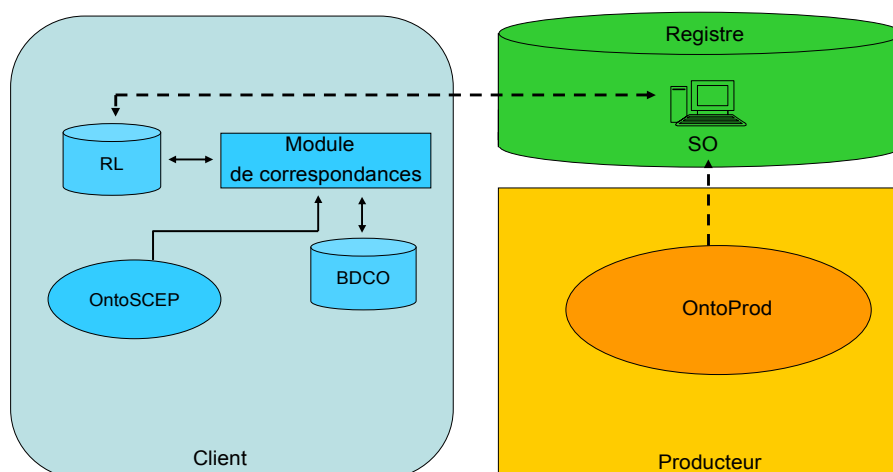


Figure III.6. SCEP-SOA intégrant les composants sémantiques ajoutés pour établir les correspondances ontologiques selon l'approche multi-ontologie

Chaque producteur décrit les services qu'il fournit selon sa propre ontologie. En publiant les services dans le registre, chaque producteur enregistre sur le serveur d'ontologies SO une copie de son ontologie. Après la découverte des services et leurs fournisseurs, le client appelle le SO pour télécharger les différentes ontologies des producteurs sélectionnés, et les stocke dans son registre local (RL). Ensuite, ces ontologies sont chargées dans le module de correspondances qui établit des correspondances bidirectionnelles entre l'ontologie locale du client et ces différentes ontologies. Ces correspondances sont ensuite stockées chez le client dans une base de données de correspondances ontologiques (BDCO).

Concernant le raisonnement sur les correspondances ontologiques établies et l'interaction avec les producteurs, les composants sémantiques ajoutés dans SCEP-SOA sont représentés sur la figure III.7.

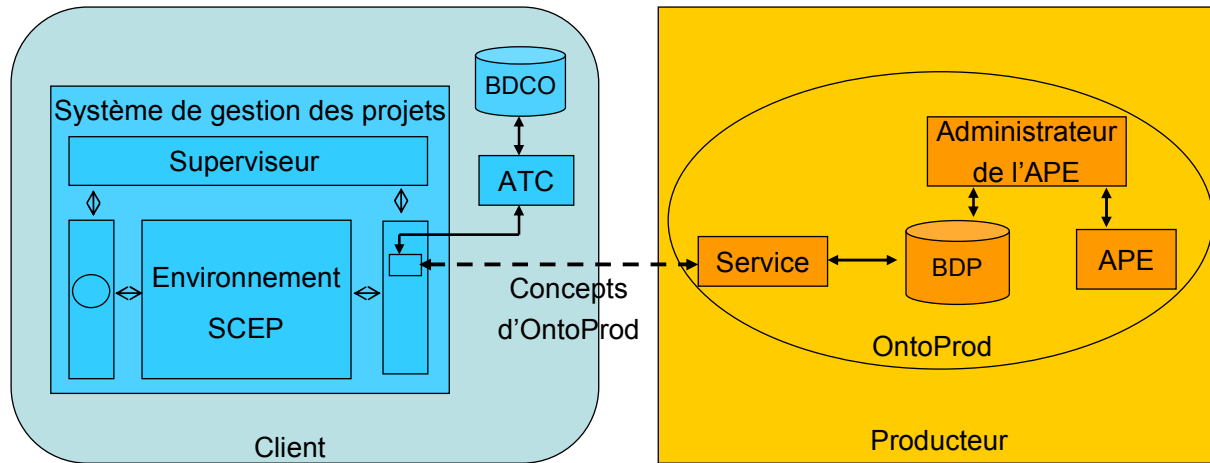


Figure III.7. SCEP-SOA intégrant les composants sémantiques pour raisonner selon l'approche multi-ontologie

Dans cette figure, seuls les composants impliqués dans l'étape d'interaction client – producteur sont présents, ce qui explique l'absence du registre qui n'intervient pas dans cette interaction. Pour simplifier la figure, un seul agent client et un seul agent ambassadeur sont présents dans le modèle SCEP chez le client.

Dans cette stratégie, la sémantique des informations échangées concerne plusieurs ontologies. En fait, avant d'invoquer le service chez le producteur correspondant, le client doit traduire les informations sur son projet de l'ontologie locale en informations décrites selon l'ontologie de ce producteur. Pour cela, un mécanisme de transformation des informations entre les ontologies est implémenté dans un agent dit agent de traduction du client (ATC). Nous remarquons l'absence de cet agent chez le producteur. Cela provient du fait que le producteur reçoit de l'agent ambassadeur correspondant les informations sur le projet client décrites selon son ontologie locale (OntoProd). Le processus d'interaction entre un client et un producteur dans SCEP-SOA intégrant le fonctionnement des composants sémantiques ajoutés dans cette stratégie est représenté sur la figure III.8.

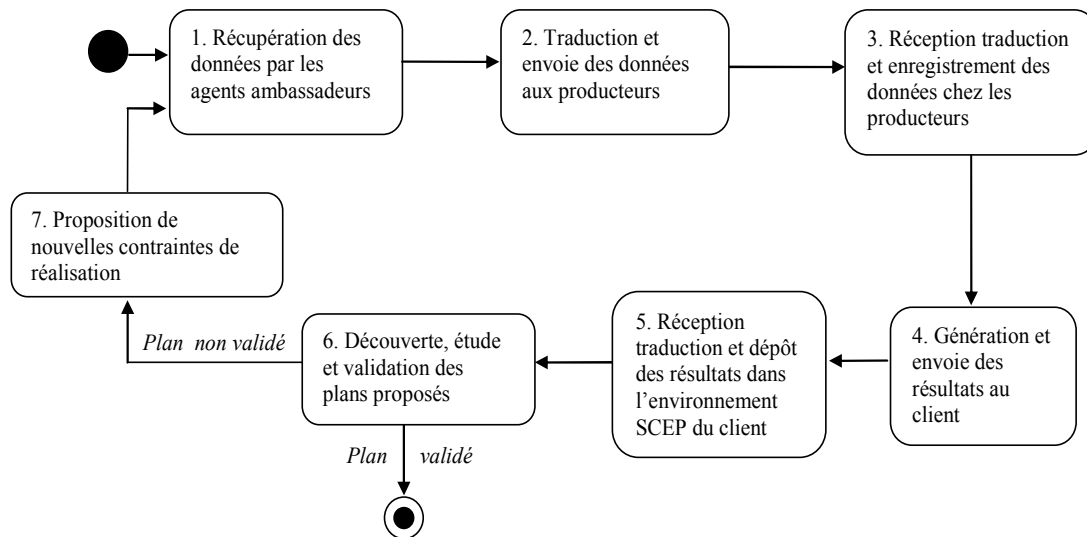


Figure III.8. Processus de gestion d'un projet client selon l'approche multi-ontologie

Ce processus est détaillé et expliqué par les diagrammes de séquences qui suivent, illustrant les différentes phases de ce nouveau fonctionnement. La collecte et la transmission des contraintes de l'étape de gestion de la réalisation des projets clients, présentée dans le chapitre précédent, sont illustrées par le diagramme de séquences présenté sur la figure III.9 montrant la prise en compte de l'aspect sémantique de cette stratégie. Notons que les étapes du traitement du projet par le producteur et de l'envoi d'un résultat au client ne sont pas concernées par cette stratégie du fait que l'aspect sémantique est à la charge du client.

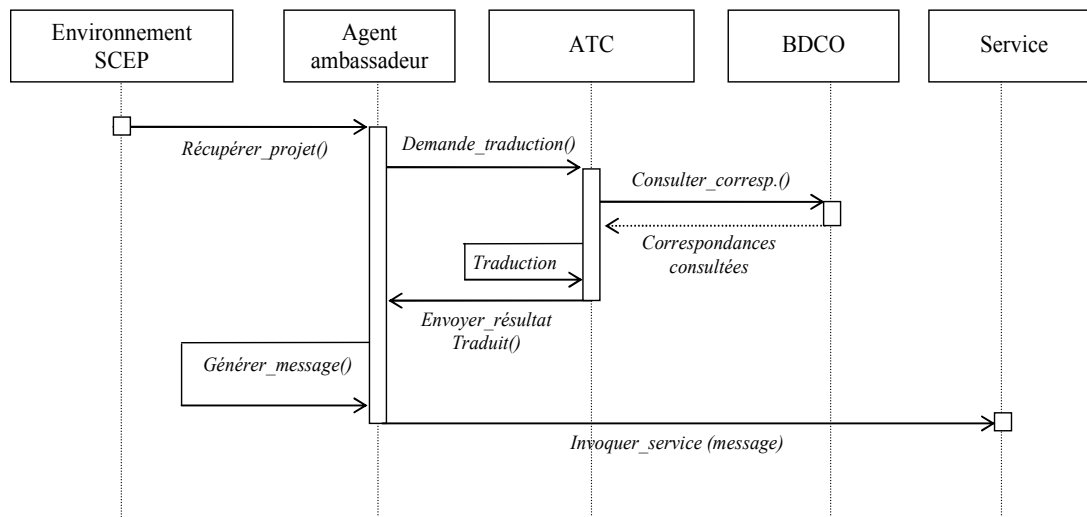


Figure III.9. Diagramme de séquences pour la collecte et la transmission des contraintes selon l'approche multi-ontologie

Après la définition du projet client selon l'ontologie locale, l'agent ambassadeur récupère de l'environnement SCEP partagé les informations sur le projet concernant le producteur avec

lequel il est chargé d'interagir. Afin de gérer les interactions avec l'ATC, l'agent ambassadeur comprend deux modules comme illustré sur la figure III.10.

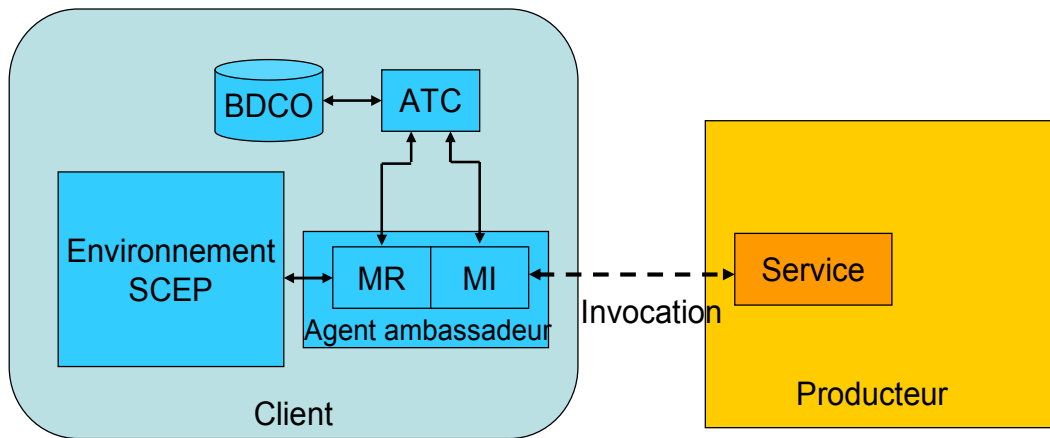


Figure III.10. Composants d'un agent ambassadeur et leurs interactions

Le module de récupération (MR) est responsable de la récupération de l'environnement de SCEP des données de la définition des projets client selon l'ontologie du client ainsi que le dépôt du résultat de l'invocation du service décrit selon cette ontologie. Le module d'invocation (MI) est responsable de l'invocation du service chez le producteur correspondant et la réception du résultat de l'invocation. Une fois les projets décrits dans l'environnement SCEP, le MR de l'agent ambassadeur récupère la description des projets demandant les services fournis par le producteur qu'il représente et l'envoie à l'agent de traduction ATC. Ce dernier traduit la description du projet de l'ontologie du client en ontologie du producteur correspondant en se basant sur les correspondances ontologiques dans la BDCO. Après la traduction, l'ATC transmet les informations traduites au MI qui récupère de ces informations les données nécessaires pour l'invocation du service chez le producteur correspondant.

La phase du traitement et validation du résultat par le client est illustrée par le diagramme de séquences présenté sur la figure III.11 montrant l'intégration de l'aspect sémantique dans cette phase.

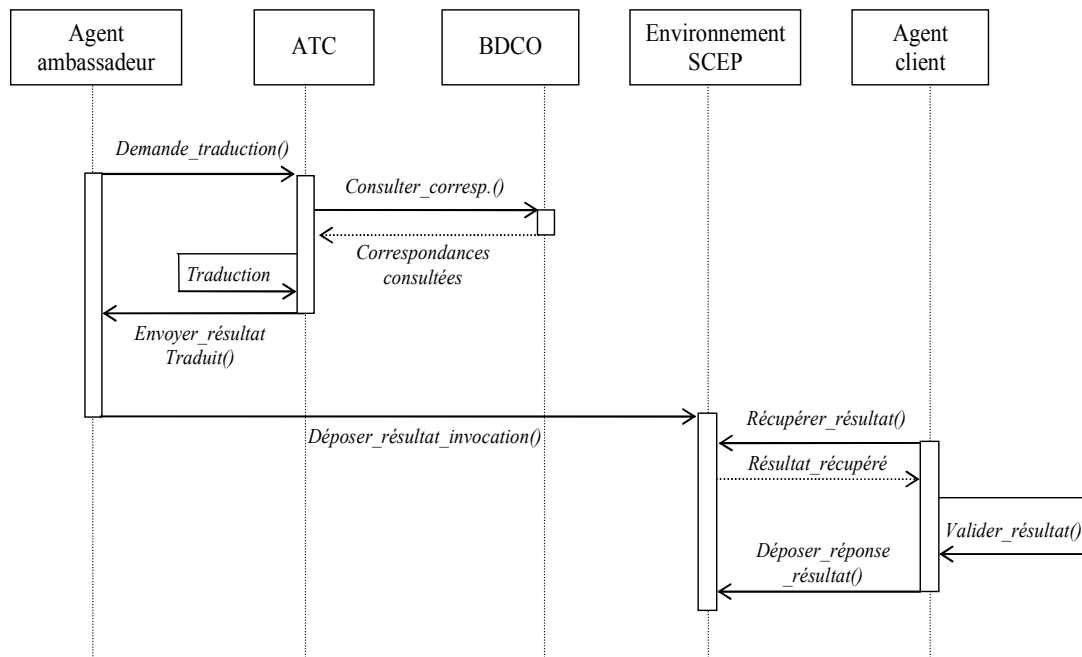


Figure III.11. Diagramme de séquences pour le traitement et la validation du résultat

Chez le producteur, en recevant les informations sur le projet client décrit selon l'ontologie de ce producteur, le service enregistre ce projet dans la base de données BDP. Après l'enregistrement du résultat de la gestion du projet établi par l'APE dans la BDP, le service récupère de cette dernière le résultat et le renvoie à l'agent ambassadeur du client. Notons que les informations envoyées par le service à l'agent ambassadeur sont décrites selon l'ontologie locale du producteur.

Chez le client, le résultat de l'invocation du service est reçu par le MI de l'agent ambassadeur qui demande à l'ATC la traduction de ce résultat en résultat décrit selon l'ontologie du client. Après traduction, l'ATC transmet au MR de l'agent ambassadeur le résultat traduit afin de le déposer dans l'environnement SCEP partagé pour le découvrir et l'interpréter par l'agent client responsable. Le résultat peut être confirmé ou validé par le client, ou bien confirmé avec modification, ou bien refusé avec modification, ou bien refusé complètement. Dans les trois premiers cas, une ré-invocation du service est nécessaire. Dans le dernier cas, le service n'est pas ré-invoqué, ce qui conduit à la suppression du résultat chez le producteur correspondant après expiration du timeout correspondant.

L'architecture globale SCEP-SOA intégrant les composants sémantiques ajoutés dans cette stratégie chez tous les acteurs est illustrée sur la figure III.12.

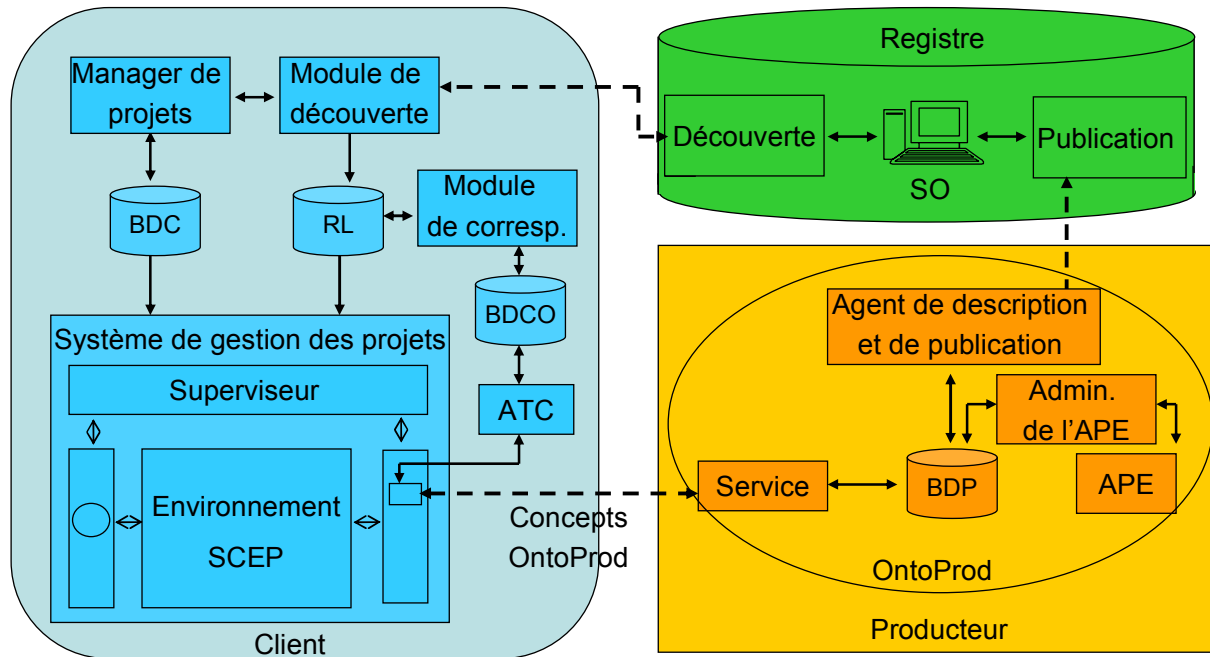


Figure III.12. SCEP-SOA intégrant tous les composants sémantiques ajoutés dans cette stratégie

7.2 Stratégie basée sur l'approche hybride

L'idée de cette stratégie est d'utiliser une ontologie commune dans l'échange des informations entre les clients et les producteurs. Pour cela, l'approche hybride a été modifiée pour mieux s'adapter au contexte de SCEP-SOA, prenant en compte l'existence préalable de différentes ontologies locales à partir desquelles une ontologie globale et commune est extraite. Cette ontologie commune comprend les concepts génériques qui peuvent être partagés entre les différentes ontologies locales. En effet, la modification de cette approche concerne l'établissement chez chaque acteur de l'architecture (client et producteur), des correspondances ontologiques entre cette ontologie commune et l'ontologie locale de l'acteur. L'approche hybride modifiée abandonne l'idée de l'approche hybride originale qui maintient l'établissement des correspondances ontologiques entre les ontologies locales. L'ontologie de base (OntoBase) est utilisée dans l'échange des informations entre le client et les producteurs. Cette idée diminue la complexité du problème et l'amène à un problème local pour chaque acteur qui consiste à établir les correspondances nécessaires entre son ontologie locale et l'ontologie globale.

Les composants sémantiques ajoutés à SCEP-SOA dans cette stratégie concernent le registre de services, le client, et le producteur. Comme dans la première stratégie, nous considérons deux étapes dans l'ajout de ces composants. La première concerne la découverte et la représentation des correspondances ontologiques. La deuxième concerne le raisonnement

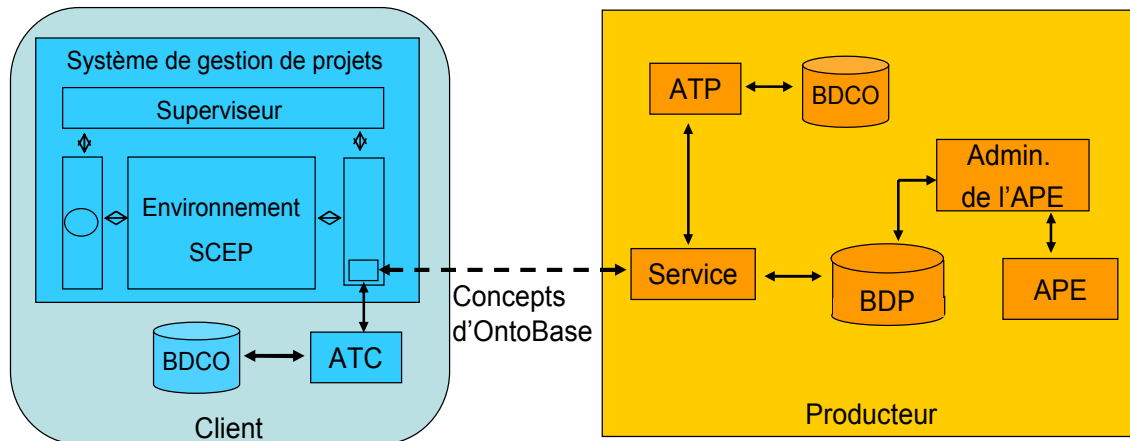


Figure III.14. SCEP-SOA intégrant les composants sémantiques ajoutés pour raisonner avec les correspondances ontologiques établies

Dans cette figure, seuls les composants impliqués dans l'interaction client – producteur sont présents. Pour simplifier la figure, nous montrons l'interaction entre un seul agent ambassadeur du client et le producteur correspondant. La composition de l'agent ambassadeur est identique à celle présentée dans la première stratégie concernant l'intégration des deux modules de récupération et d'invocation (MR et MI). Au niveau de chaque acteur (client ou producteur), un mécanisme de raisonnement sur les correspondances ontologiques établies est mis en place.

Chez le client, un agent de traduction noté ATC est chargé de traduire les informations ou données décrivant le projet client selon l'ontologie du client en données décrites selon l'ontologie globale OntoBase. Il traduit aussi les informations contenues dans le résultat reçu de l'invocation du service décrit selon OntoBase en informations décrites selon l'ontologie locale du client.

Chez le producteur, un agent de traduction noté ATP est chargé de la traduction des informations contenues dans la requête d'invocation du service, décrites selon OntoBase et qui concernent la description du projet client. Il traduit aussi les informations contenues dans le résultat de l'invocation, décrites selon l'ontologie locale du producteur (OntoProd) en informations décrites selon OntoBase afin d'être envoyées au client correspondant. La figure III.14 présentée ci-dessus ne montre pas les liens entre les ontologies présentes chez le producteur, *i.e.* OntoBase et OntoProd, et les autres composants, notamment la BDP et le service. La figure III.15 détaille la structure de ces composants, montrant leur considération par ces deux ontologies.

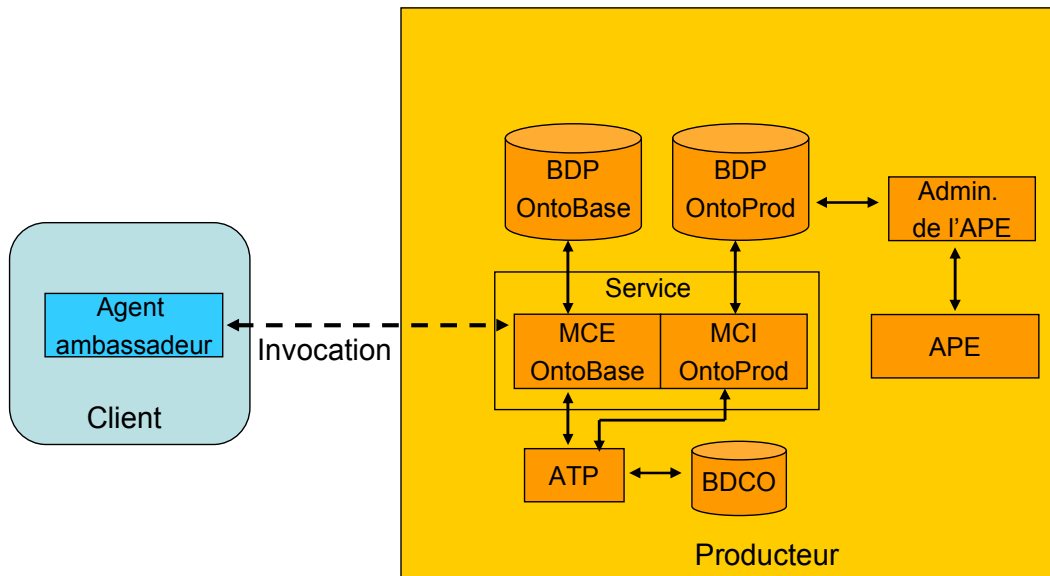


Figure III.15. Structure du service et de la BDP chez le producteur

En effet, le service comprend deux modules, un module de communication externe (MCE) qui est chargé de l'interaction avec les clients, et un module de communication interne (MCI) chargé de l'interaction avec les composants internes. MCE manipule des concepts de l'ontologie globale OntoBase tandis que MCI manipule des concepts de l'ontologie locale OntoProd. Concernant la BDP, elle est mise en place en deux versions, une pour OntoBase qui interagit avec le MCE du service, et une pour OntoProd qui interagit avec le MCI du service et avec l'application de gestion APE. La BDP OntoBase stocke des informations sur les projets clients décrits selon OntoBase telles qu'elles sont dans les requêtes d'invocation du service. Elle stocke aussi les résultats de gestion des projets établis par l'APE et traduits en OntoBase. La BDP OntoProd stocke des informations sur les projets clients traduits en OntoProd. Elle stocke aussi les résultats établis par l'APE, décrits en OntoProd. Le processus de fonctionnement de ces différents composants et modules est décrit dans la suite.

Après l'ajout des composants sémantiques, les activités traitées dans le nouveau fonctionnement de SCEP-SOA concernant l'étape de la gestion de la réalisation d'un projet client sont résumées par le processus présenté sur la figure III.16.

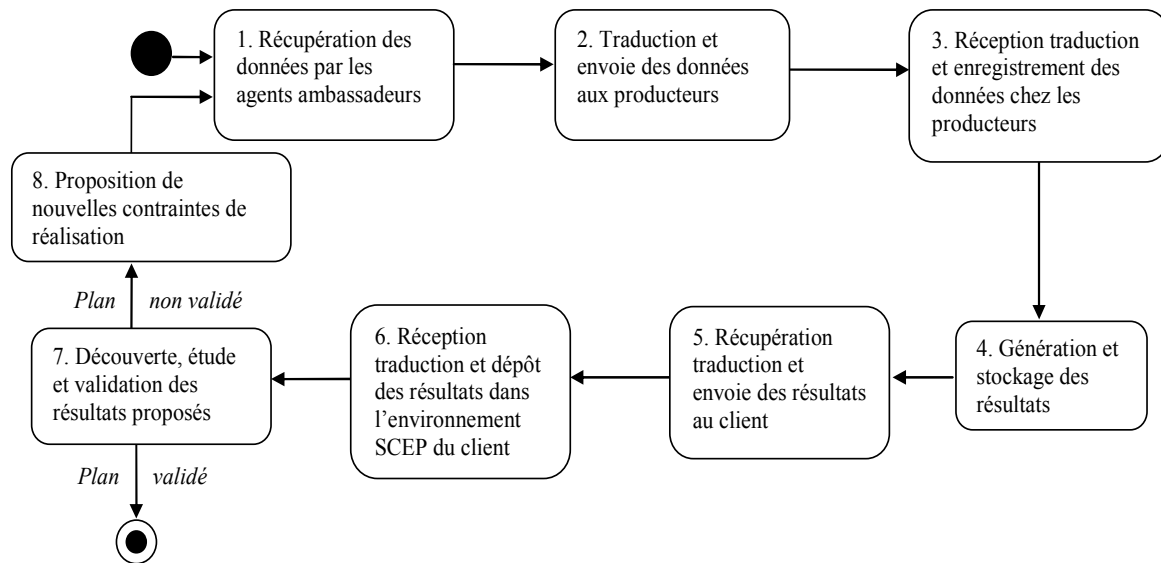


Figure III.16. Processus de gestion d'un projet multi-site

Concernant la collecte et la transmission des contraintes, les séquences sont identiques à celles présentées dans le diagramme de séquences de la première stratégie présentée sur la figure III.9. En fait, le module de récupération MR de l'agent ambassadeur récupère les informations sur le projet client de l'environnement SCEP. Cet agent demande ensuite à l'ATC de traduire le projet client décrit selon l'ontologie locale en un projet décrit selon l'ontologie OntoBase afin de l'envoyer au producteur correspondant. Pour traduire les informations, l'ATC se base sur les correspondances ontologiques stockées dans la BDCO, et renvoie au module d'invocation MI de l'agent ambassadeur la traduction du projet client. Une fois le projet traduit reçu par le MI, il invoque le service chez le producteur correspondant en lui transmettant les informations nécessaires décrites selon l'ontologie OntoBase.

Les phases de traitement du projet par le producteur et l'envoi du résultat au client sont illustrées par le diagramme de séquences représenté sur la figure III.17.

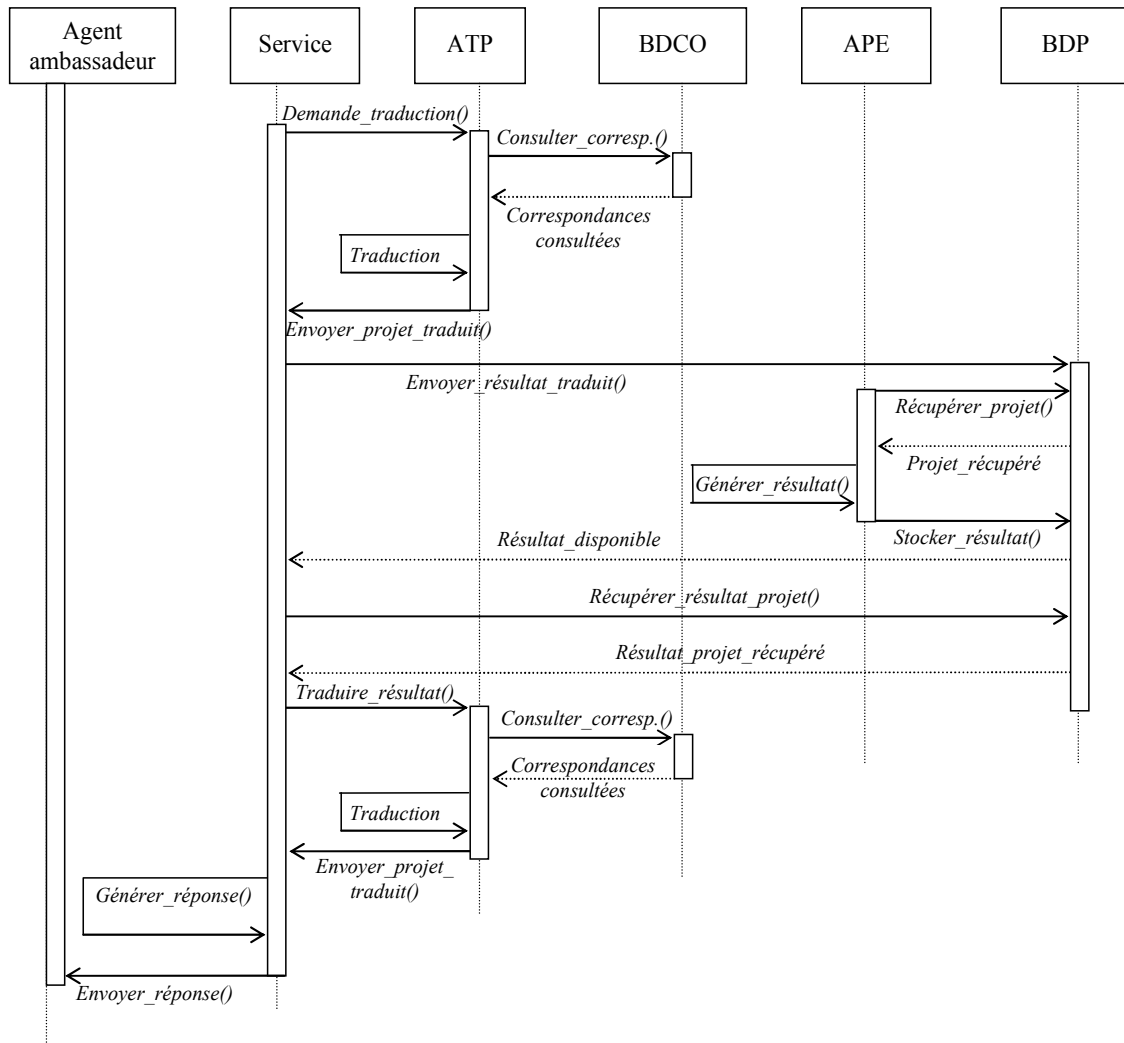


Figure III.17. Diagramme de séquences pour le traitement du projet et l'envoi du résultat

A la réception de la requête d'invocation du service chez le producteur, le module MCE du service enregistre les informations sur le projet client, contenues dans cette requête et décrites selon OntoBase, dans la BDP OntoBase, et demande ensuite à l'ATP de traduire ces informations. Comme chez le client, l'ATP s'appuie sur les correspondances ontologiques stockées dans la BDCO pour traduire ces informations en informations décrites selon l'ontologie locale du producteur (OntoProd). La traduction est renvoyé ensuite par l'ATP au module MCI du service qui l'enregistre dans la BDP OntoProd. Après le lancement de l'APE, le résultat établi et décrit selon OntoProd est stocké dans la BDP OntoProd. Ensuite, le module MCI du service récupère ce résultat de la BDP OntoProd et demande à l'ATP de le traduire en résultat décrit selon l'ontologie globale OntoBase afin d'être envoyé au client correspondant. Après traduction, l'ATP renvoie au module MCE du service le résultat traduit. Ce résultat est ensuite stocké par le module MCE du service dans la BDP OntoBase, puis envoyé par le MCE à l'agent ambassadeur correspondant.

Chez le client, une fois le résultat de l'invocation du service reçu par le module MI de l'agent ambassadeur, ce dernier contacte l'ATC pour traduire le résultat proposé par le producteur pour ce projet client et décrit selon l'ontologie OntoBase en un résultat décrit selon l'ontologie locale du client. En recevant de l'ATC le résultat de l'invocation traduit, le module MR de l'agent ambassadeur dépose les informations contenues dans ce résultat dans l'environnement partagé de SCEP pour être bien interprété et étudié par l'agent client responsable. Le diagramme de séquences correspondant à cette phase est identique à celui présenté sur la figure III.11 dans la première stratégie.

L'architecture globale de SCEP-SOA intégrant les composants sémantiques ajoutés dans cette stratégie chez tous les acteurs est illustrée sur la figure III.18.

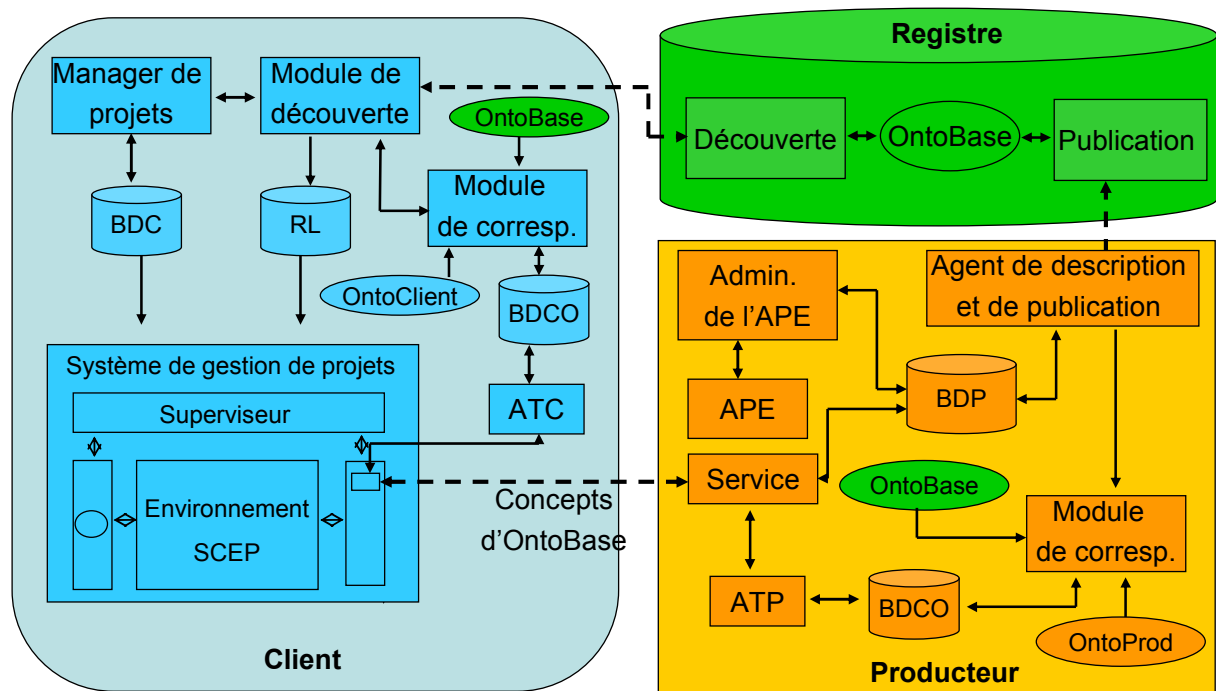


Figure III.18. SCEP-SOA intégrant tous les composants sémantiques selon cette stratégie

8 Comparaison des stratégies étudiées

La comparaison entre ces deux stratégies dans le contexte de l'architecture SCEP-SOA est présentée dans la Table III.1. Elle illustre les avantages et les limites de ces stratégies selon les critères considérés.

Stratégie <i>Critère</i>	Basée sur l'approche multi-ontologie	Basée sur l'approche hybride modifiée
<i>Effort d'implémentation</i>	Couteuse pour les clients	Raisnable pour les clients et les producteurs
<i>Mécanismes de translation</i>	Complexe chez les clients	Moins complexe chez les clients et les producteurs
<i>Ajout/suppression d'ontologies</i>	Besoin de modification des correspondances ontologiques	Pas d'influence sur les correspondances ontologiques
<i>Perte d'information</i>	Très probable	Moins probable
<i>Convergence sémantique des informations échangées</i>	Pas nécessairement assurée	Assurée grâce à OntoBase
<i>Approche d'interopérabilité du FEI (CEN/ISO 11354)</i>	Fédérée	Unifiée

Table III.1. Comparaison des deux stratégies dans SCEP-SOA

Dans la première stratégie basée sur l'approche multi-ontologie, chaque producteur garde sa propre ontologie locale et reçoit des données décrites selon cette ontologie. Cela n'implique aucune modification sémantique chez les producteurs, ce qui réduit les coûts de mise en place de cette stratégie chez les producteurs, notamment les PME ayant des moyens limités. Aussi, du fait qu'aucun mécanisme de traduction n'est nécessaire chez un producteur, cela réduit le temps de traitement du service chez cet acteur ainsi que le temps d'attente du résultat d'une invocation de service du côté du client. Selon le référentiel FEI (Framework for Enterprise Interoperability) proposé par Chen *et al.*, (2006), cette stratégie peut être considérée comme une approche fédérée pour l'interopérabilité sémantique du fait qu'aucune ontologie ou vocabulaire communs sont utilisés. Les correspondances entre les ontologies sont établies d'une manière directe sans utilisation d'une ontologie de référence prédéterminée.

Néanmoins, du fait que les correspondances ontologiques établies par le client dans cette stratégie concernent d'une part l'ontologie du client et d'autre part les ontologies des différents producteurs partenaires, cela augmente d'une manière importante le nombre de ces correspondances. Ce nombre dépend du nombre des producteurs ou plus précisément du nombre des ontologies mises en jeu, ainsi que du nombre de concepts définis dans chacune de ces ontologies. Ce nombre important de correspondances nécessite une taille importante de la base de données pour stocker ces correspondances et augmente la complexité des mécanismes

de traduction et de raisonnement avec ces correspondances, rendant la mise en place de cette stratégie très coûteuse chez les clients. En outre, le client doit établir des correspondances ontologiques avec les ontologies de tous les producteurs, même si après interaction, certains d'entre eux ne sont pas retenus pour la réalisation des projets clients. Cela illustre aussi le coût important de l'application de cette stratégie chez le client. Le déploiement de cette stratégie nécessite par ailleurs des compétences techniques élevées pour pouvoir raisonner avec les correspondances ontologiques afin de garantir la conservation de la sémantique des informations échangées. Cela nécessite l'intervention d'experts ontologiques (Dou *et al.*, 2005), ce qui implique un coût technique important de mise en place de cette stratégie.

Dans cette stratégie, les correspondances ontologiques établies chez le client dépendent fortement de la cohérence entre les versions des ontologies des producteurs avec lesquels ce client interagit. Chaque modification dans l'ontologie d'un producteur, ou l'ajout d'un nouveau producteur donc d'une nouvelle ontologie, entraîne des mises à jour au niveau des correspondances ontologiques établies chez le(s) client(s) concerné(s). En outre, l'hétérogénéité structurelle qui peut exister entre ces ontologies met en évidence la non existence de certains concepts dans certaines ontologies, affectent ainsi la cohérence des informations échangées en augmentant le risque de perte d'informations. Un exemple de cette perte est illustré par le concept de coût de réalisation de l'activité de production, défini dans l'ontologie du client et qui n'est pas pris en compte dans d'autres ontologies, *i.e.* ontologie du modèle TAPAS (Moutarlier *et al.*, 2000). En outre, du fait de l'absence d'un vocabulaire commun et partagé, la convergence sémantique n'est pas assurée entre les partenaires dans cette première stratégie.

Dans la deuxième stratégie, l'application de l'approche hybride modifiée réduit les correspondances ontologiques qui se limitent à des correspondances avec une seule ontologie qui est l'ontologie globale (OntoBase). Les correspondances ontologiques établies chez le client s'effectuent sans tenir compte des particularités des ontologies des producteurs comme dans la première stratégie. Cela réduit les coûts de mise en place de cette stratégie chez les clients ainsi que la complexité des mécanismes de traduction entre les ontologies. Aussi, cette stratégie offre une autonomie sémantique entre les partenaires où chacun peut modifier son ontologie locale sans influencer sur les correspondances ontologiques chez les autres partenaires, *i.e.* les clients. Donc, l'ajout et la suppression des producteurs n'influencent pas

sur les correspondances ontologiques déjà établies du fait que ces dernières sont indépendantes de toute ontologie d'un producteur.

Un autre point important dans cette deuxième stratégie consiste en l'utilisation d'une ontologie commune et globale dans l'échange d'informations entre les partenaires. Ceci assure une convergence sémantique des différentes ontologies utilisées par les partenaires vers cette ontologie globale ainsi qu'une standardisation des informations échangées. Cette convergence sémantique garantit une bonne interprétation des informations échangées entre les différents partenaires. La présence de cette ontologie globale réduit le risque de perte d'informations et permet une meilleure conservation sémantique des informations échangées. Par exemple, si un concept de l'ontologie du client n'est pas pris en compte par une ontologie d'un producteur, l'information décrite par ce concept reste présente durant l'échange d'informations entre les partenaires. En effet, l'information décrite par ce concept est stockée dans la BDP OntoBase du producteur bien qu'elle ne soit pas interprétée par l'APE de ce dernier. Cette information est rajoutée au résultat traduit en OntoBase avant de l'envoyer au client correspondant. Cela illustre la conservation sémantique des informations échangées et permet à ce producteur, si les contraintes modélisées par les concepts communs contenues dans le résultat renvoyé par ce producteur peuvent être validées par le client, de ne pas être pénalisé par le client dans le choix de ses partenaires.

Dans cette stratégie, l'extension de l'ontologie globale est possible pour l'enrichir par d'autres concepts, ce qui peut améliorer et faciliter l'établissement des correspondances ontologiques chez certains acteurs ainsi que le raisonnement avec ces correspondances. Cependant, cela peut avoir de mauvaises conséquences du fait que les nouveaux concepts ajoutés à l'ontologie globale peuvent entraîner des mises à jour importantes dans les correspondances ontologiques établies par les différents acteurs.

Selon le référentiel FEI, cette stratégie peut être considérée comme une approche unifiée pour l'interopérabilité sémantique en raison de la présence de l'ontologie globale et commune comme un modèle de référence prédéterminé et une référence sémantique pour les concepts échangés entre les partenaires. Cette ontologie globale est considérée comme un format commun à un méta-niveau et une base d'équivalence sémantique pour établir des correspondances entre différents méta-modèles représentés par les différentes ontologies des partenaires.

Les points forts présentés ci-dessus favorisent l'adoption de cette deuxième stratégie dans note architecture SCEP-SOA pour assurer l'interopérabilité sémantique et garantir la bonne compréhension et interprétation des informations échangées entre les différentes applications de production utilisées par les partenaires.

9 Conclusion

Les ontologies sont actuellement en cours d'application dans la majorité des domaines industriels et commerciaux nécessitant un échange électronique des données entre des systèmes et applications hétérogènes. Dans ce chapitre, nous avons étudié les ontologies comme outils d'interopérabilité sémantique pour garantir la bonne compréhension des informations échangées entre les partenaires afin de les intégrer dans notre architecture SCEP-SOA. L'hétérogénéité entre les ontologies est un problème important dans cette architecture du fait que les partenaires n'ont pas nécessairement la même connaissance du domaine de la production et utilisent de ce fait des vocabulaires différents pour modéliser les concepts utilisés par leurs systèmes et leurs applications.

Bien que l'approche multi-ontologie appliquée dans la première stratégie permette de résoudre le problème d'hétérogénéité entre les ontologies, son application dans le contexte de SCEP-SOA est complexe et très coûteuse. Ceci est dû à la multiplicité des correspondances ontologiques à établir et de la complexité des mécanismes de raisonnement avec ces correspondances pour la traduction des informations entre les différentes ontologies. L'application de l'approche hybride modifiée dans la deuxième stratégie semble être plus convenable à notre contexte du fait qu'elle réduise la complexité de ces correspondances et qu'elle assure une convergence et une conservation de la sémantique des informations échangées entre les partenaires. Afin d'illustrer la faisabilité de l'application de cette stratégie d'interopérabilité sémantique dans l'architecture SCEP-SOA proposée, l'implémentation d'un cas d'étude a été réalisée dans le contexte de la planification de projets de production multi-sites. Cette implémentation fait l'objet du prochain chapitre.

Chapitre IV

Application de SCEP-SOA à l'interopérabilité des systèmes de planification hétérogènes

1 Introduction

Plusieurs travaux se sont intéressés dans la littérature à la problématique de la gestion de la planification multi-site. Des modèles basés sur les systèmes multi-agents ont été proposés pour cet objectif (Frankovic *et al.*, 2002 ; Lima *et al.*, 2006 ; Wong *et al.*, 2006 ; Nishioka, 2004 ; Lazansky *et al.*, 2001 ; Thibaud *et al.*, 2007). Cependant, ces travaux s'intéressent très peu à la problématique de l'interopérabilité, et notamment l'interopérabilité sémantique, entre les systèmes de planification hétérogènes qui constitue l'objectif principal de SCEP-SOA.

L'objectif de ce chapitre est de montrer la faisabilité de la stratégie d'interopérabilité sémantique du modèle SCEP-SOA. Pour cela, un cas d'étude est présenté sur l'interopérabilité sémantique entre des systèmes de planification et d'ordonnancement hétérogènes. La contribution apportée dans ce chapitre n'est pas liée à la résolution du problème d'ordonnancement ou de planification multi-site mais à la possibilité de faire interopérer, au niveau sémantique, différentes applications hétérogènes de planification. Aussi, les outils de planification considérés dans notre étude sont contrôlables, *i.e.* qu'ils peuvent être pilotés manuellement (lancement, re-planification, etc.).

Le logiciel R@mses apporte une solution satisfaisante pour la réalisation d'ordonnancements distribués ou multi-sites dans le cadre de systèmes homogènes. La convergence de la méthode de résolution basée sur l'utilisation du modèle SCEP a été démontrée dans ce contexte (Archimède et Coudert, 2001b). L'objectif ici est donc de montrer l'interopérabilité sémantique dans l'échange d'informations entre des outils de planification hétérogènes. Aussi, étant donné que l'élaboration d'un planning requiert un nombre fini de cycles identiques du point de vue fonctionnel, nous nous limitons dans le cas étudié, à l'étude de l'échange d'informations pour un seul cycle de planification et notamment ceux concernant les fonctionnalités d'invocation du service par le client et du traitement du résultat obtenu.

La description du cas d’étude est présentée dans la section 2. Les outils de planification utilisés par les partenaires sont décrits dans la section 3 qui présente aussi l’ontologie de planification utilisée par chaque système ainsi que l’hétérogénéité sémantique entre ces différentes ontologies. La section 4 décrit l’ontologie commune de planification proposée qui servira dans l’annotation des informations échangées entre les différents systèmes de planification. La section 5 définit les modèles de la requête et de la réponse d’invocation du service de planification. L’étude des interactions entre les partenaires dans un cycle de planification, présentée dans la section 6, illustre l’interopérabilité sémantique assurée par la stratégie sémantique hybride adoptée.

2 Description générale du cas d’étude traité concernant deux producteurs et un client ayant deux projets

Nous considérons un marché orienté-service dédié au domaine de la fabrication. Les services proposés permettent de planifier des activités de fabrication, telles que le sciage, le polissage, le perçage, etc. Ces activités sont réalisées par deux producteurs répartis géographiquement sur deux sites différents dont la configuration est décrite sur le Tableau IV.1.

Site	Machine	Activité
Producteur 1	M1	Sciage verre
		Polissage verre
	M2	Sciage bois
		Polissage bois
Producteur 2	M3	Sciage verre
		Sciage bois
	M4	Perçage verre
		Perçage bois
	M5	Assemblage

Tableau IV.1. Machines et activités chez les producteurs

Les activités de Sciage (bois et verre) sont proposées par les deux producteurs. Par ailleurs, le producteur 1 propose des activités de Polissage (bois et verre). Le producteur 2 offre, outre les activités de sciage (bois et verre), des activités d’Assemblage et de Perçage (bois et verre). Comme illustré sur le Tableau IV.1, une machine peut réaliser une ou plusieurs activités. Les activités de sciage verre et sciage bois sont réalisées sur la même machine (M3) chez le

producteur 2. Elles sont réalisées sur deux machines distinctes (M1 et M2) chez le producteur 1.

Nous considérons une entreprise cliente désirant réaliser deux projets de production correspondant chacun à la fabrication d'un produit. Ces projets sont multi-sites du fait que leur réalisation nécessite la participation des deux producteurs. La planification de ces projets est réalisée d'une manière distribuée par coopération entre les systèmes de planification utilisés par les différents partenaires. Dans notre étude, nous ne prenons pas en compte le transport inter-site du fait que nous nous intéressons plus à l'aspect hétérogénéité sémantique des systèmes de planification des partenaires dans la gestion des projets multi-sites plutôt qu'à l'aspect synchronisation et gestion du transport inter-site. Le transport intra-site est pris en compte localement par chaque producteur.

Les projets du client doivent être décrits en fonction des activités du marché. Par souci de simplification, nous ne nous intéressons pas aux étapes de publication et de découverte des services. Nous considérons que les activités demandées par le client sont publiées dans le registre global, et que les services correspondants ont été découverts par le client qui a défini ses projets en fonction des activités fournies par ces services.

Les projets du client correspondent respectivement à la fabrication d'une chaise et d'une table basse. La chaise est constituée de quatre pieds en bois, d'une assise en bois et d'un dossier en verre. La table est constituée de quatre pieds en bois et d'un plateau en verre. Quatre ordres de fabrication (OF) sont nécessaires pour la fabrication de la chaise et trois OF pour la table. Les gammes de fabrication de ces projets sont décrites sur les Tableaux IV.2 et IV.3 respectivement.

OF	Gamme	Activité	Durée
OF1	(Préparation pieds)	Sciage bois	20 min
		Polissage bois	20 min
		Perçage bois	10 min
OF2	(Préparation dossier)	Sciage verre	20 min
		Perçage verre	10 min
		Polissage verre	10 min
OF3	(Préparation assise)	Sciage bois	20 min
		Polissage bois	20 min
		Perçage bois	10 min
OF4	(Assemblage final)	Assemblage	20 min

Tableau IV.2. Description des ordres de fabrication de la chaise (projet 1)

OF	Gamme	Activité	Durée
OF1	(Préparation pieds)	Sciage bois	20 min
		Polissage bois	20 min
		Perçage bois	10 min
OF2	(Préparation plateau)	Sciage verre	20 min
		Perçage verre	10 min
		Polissage verre	10 min
OF3	(Assemblage final)	Assemblage	20 min

Tableau IV.3. Description des ordres de fabrication de la table (projet 2)

La fabrication d’une chaise ainsi que celle de la table nécessitent que toutes les pièces convergent en fin de fabrication vers le site du producteur 2 pour l’assemblage final. La réalisation des ordres de fabrication OF1, OF2 et OF3 précède celle de l’OF4 dans le projet 1 et la réalisation des OF 1 et 2 doit être effectuée avant l’OF 3 dans le projet 2.

3 Outils de planification utilisés par les partenaires

Le cas étudié concerne l’interaction entre un client disposant d’un système de planification R@mses (Re@ctive Multi-site SystEm for Scheduling) (Archimède *et al.*, 2003) basé sur le modèle SCEP, et deux producteurs. Le producteur 1 dispose du même système que le client, et le deuxième utilise un système de planification basé sur le modèle TAPAS (The Almost Perfect Approach to Scheduling) (Moutarlier *et al.*, 2000). Le fonctionnement et le méta-

modèle de TAPAS différent de celui de SCEP. La conception de ces deux modèles a pourtant été réalisée au sein du Laboratoire Génie de Production de l'Ecole Nationale d'Ingénieurs de Tarbes par des concepteurs différents. Bien que ces systèmes aient le même objectif fonctionnel, ils ne manipulent pas les mêmes concepts de planification. Cela illustre l'hétérogénéité sémantique entre ces systèmes qui est détaillée dans l'étude de leur ontologie de planification.

3.1 Présentation de l'ontologie de R@mses

L'ontologie utilisée par le système R@mses notée R@msesOntology comprend 8 classes qui sont représentées sur la figure IV.1.

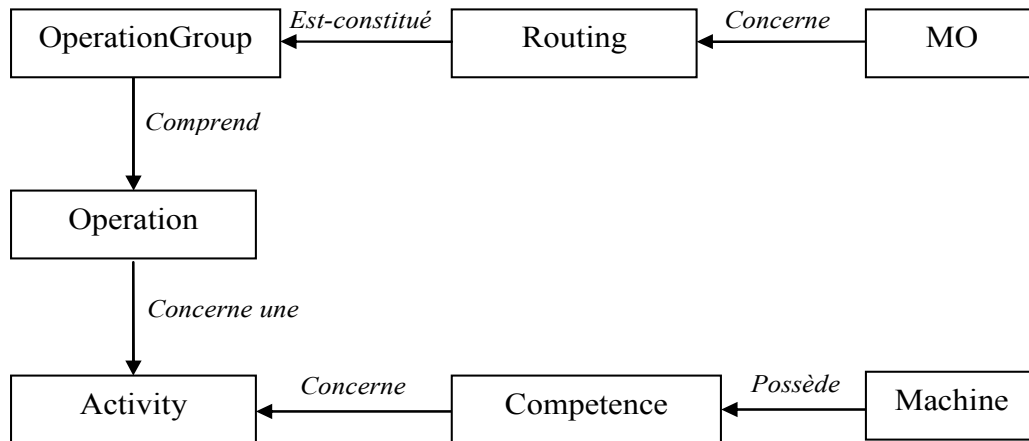


Figure IV.1. Cadre conceptuel de l'ontologie de R@mses (R@msesOntology)

Dans cette ontologie, un ordre de fabrication (MO) concerne une seule gamme (Routing) séquentielle de groupes d'opérations (OperationGroup). Un OperationGroup comprend plusieurs opérations qui peuvent être séquentielles, parallèles, ou indifférents indiquant qu'il n'y a aucune relation reliant les opérations entre elles, etc. Une Operation concerne une seule activité (Activity). Une Machine peut posséder plusieurs compétences (Compétence). Une Compétence concerne une activité et décrit les contraintes sur les machines pour la réalisation des activités *i.e.* coût de réalisation de l'activité et la capacité de la machine, etc.

Nous considérons que toutes les activités d'un projet sont interruptibles et non transférables et que leur temps de préparation (Setup Time) est nul. Elles ont toutes le même coût de réalisation souhaité. La gamme de chaque MO comprend un seul groupe d'opérations séquentielles regroupant les activités à l'exception de l'OF2 du projet 1 qui en comprend

deux : le premier groupe comprenant les activités de Sciage, Polissage et Perçage de verre, le deuxième les activités de Gravure et Teinture. Ces groupes d'opérations sont séquentiels.

3.2 Présentation de l'ontologie de TAPAS

Le modèle TAPAS (Moutarlier *et al.*, 2000) est un cadre modulaire pour l'ordonnancement supportant le développement des ordonnanceurs mieux adaptés aux workshop dans lesquels ils sont utilisés. Ce modèle manipule des concepts définis dans l'ontologie de TAPAS notée TAPASontology. Cette ontologie comprend six classes et sous classes qui sont représentées sur la figure IV.2.

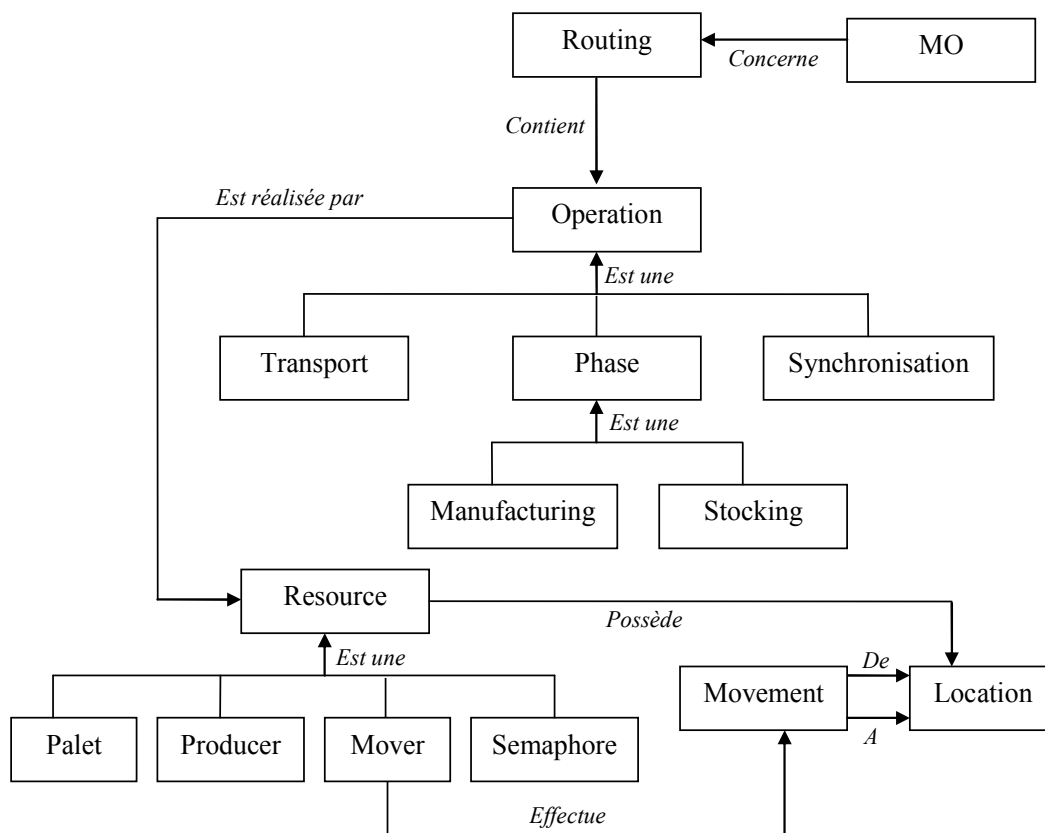


Figure IV.2. Cadre conceptuel de l'ontologie de TAPAS (TAPASontology)

Dans cette ontologie, un ordre de fabrication (MO) concerne une seule gamme (Routing) qui peut contenir plusieurs opérations (Operation). La classe Operation comprend trois sous-classes indiquant le type de l'opération : Transport, Phase, ou Synchronisation. La phase elle-même comprend deux sous-classes indiquant le type de la phase : fabrication (Manufacturing) ou stockage (Stocking). Une Operation est réalisée par une ressource (Resource) qui comprend quatre sous-classes indiquant le type de la ressource : Palet, Producer, Mover, ou Semaphore. Le type Producer modélise une machine de fabrication tandis que le type Mover

modélise le transport intra-site. Une ressource peut posséder plusieurs locations. Une ressource de type Mover effectue des mouvements (Movement) entre les locations afin de transporter les pièces entre les machines.

Dans notre cas d'étude, nous considérons que les temps de préparation des opérations sont nuls.

3.3 Hétérogénéités entre R@msesOntology et TAPASontology

Plusieurs hétérogénéités existent entre les ontologies TAPASontology et R@msesOntology conduisant à un problème d'hétérogénéité sémantique lié aux différences qui peuvent exister dans la compréhension et l'interprétation des concepts de planification échangés entre les applications.

L'hétérogénéité sémantique entre TAPASontology et R@msesOntology provient du fait que ces ontologies n'utilisent pas les mêmes concepts et vocabulaires de planification. Elle est illustrée par les hétérogénéités structurelles et terminologiques. L'hétérogénéité terminologique est illustrée par les différents concepts utilisés pour représenter le même concept de planification. Nous citons par exemple la classe *Producer* dans TAPASontology qui est représentée par la terminologie *Machine* dans R@msesOntology. L'hétérogénéité qui est plus présente dans cet exemple est l'hétérogénéité structurelle. Cette hétérogénéité peut être clairement remarquée en comparant la structure des classes des deux ontologies représentée sur la figure IV.3.

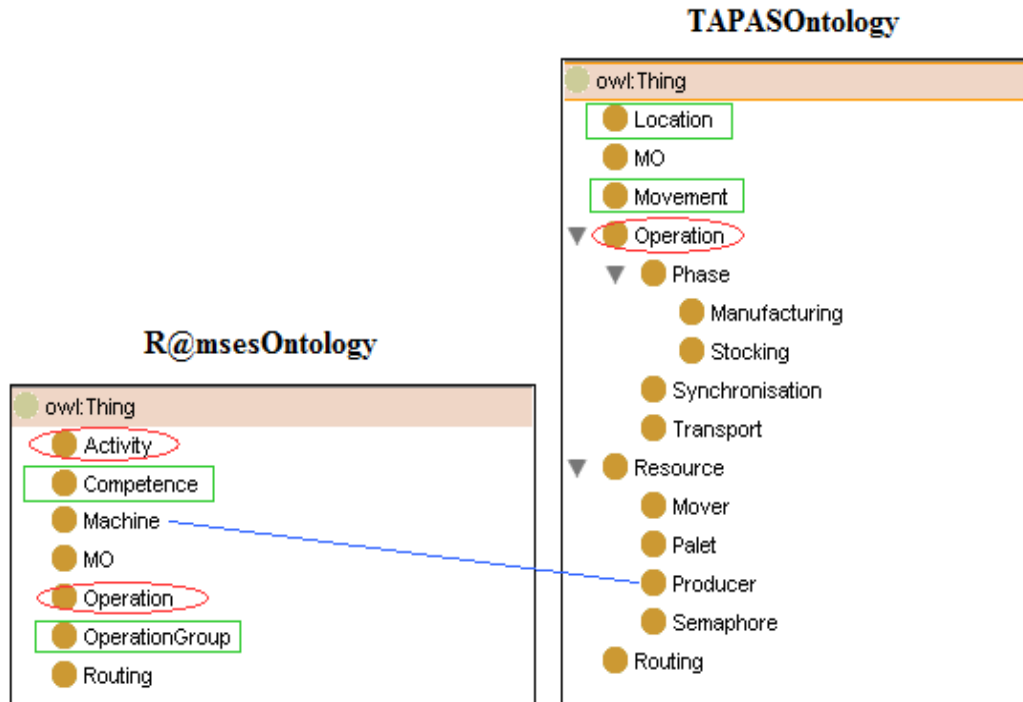


Figure IV.3. Hétérogénéités structurelle et terminologique entre les ontologies de R@mses et de TAPAS

L'hétérogénéité structurelle est présente entre les classes *Activity* et *Operation* de l'ontologie R@msesOntology et la classe *Operation* de l'ontologie TAPASontology, entourées en rouge. En effet, les deux classes (*Activity* et *Operation*) de R@msesOntology sont structurées en une seule classe dans TAPASontology, qui est la classe *Operation* ou plus précisément la sous-classe *Manufacturing*. Outre cet exemple de généralisation-spécialisation, l'hétérogénéité structurelle entre ces deux ontologies est aussi illustrée par la présence de certains concepts qui sont absents dans l'autre, *i.e.* les classes entourées en vert sur la figure IV.3. La classe *OperationGroup* de l'ontologie R@msesOntology modélise les liens temporels entre les opérations. Cela est modélisé par les propriétés de la classe *Manufacturing* de l'ontologie TAPASontology, à savoir les propriétés d'objets *hasNextOperation* et *hasPreviousOperation*.

L'hétérogénéité structurelle ne concerne pas seulement les classes des ontologies mais aussi les propriétés de ces classes où certaines propriétés présentes dans l'ontologie de SCEP ne sont pas représentées dans l'ontologie de TAPAS et *vice versa*, *i.e.* propriétés *Transferable* et *Interruptible* dans R@msesOntology absentes dans TAPASontology, etc. Néanmoins, dans le cas étudié, nous ne nous intéressons pas à cette différence du fait qu'on considère que les opérations sont toutes interruptibles et non transférables.

L'hétérogénéité terminologique est illustrée entre la classe *Machine* dans R@msesOntology et *Producer* dans TAPASontology. Ces différentes terminologies expriment le même concept de ressource de fabrication. L'hétérogénéité terminologique ne se limite pas seulement aux classes mais aussi à leurs propriétés, *i.e.* *PrepTime* et *ProdTime* dans TAPASontology qui correspondent respectivement aux propriétés *SetupTime* et *OperatingTime* dans R@msesOntology. Ces hétérogénéités sont gérées par la mise en place d'une ontologie globale et commune pour la planification, utilisée dans l'échange des informations entre les partenaires. Cette ontologie, notée OntoBase, est définie et étudiée dans la section suivante.

4 L'ontologie globale OntoBase

Pour gérer cette hétérogénéité sémantique, nous avons mis en place l'ontologie globale de planification « OntoBase ». Elle assure l'interopérabilité entre les différentes applications de planification présentes chez les partenaires. Elle est basée sur l'ontologie d'ordonnancement OZONE proposée par Smith et Becker (Smith et Becker, 1997) dont les concepts de base sont représentés sur la figure IV.4.

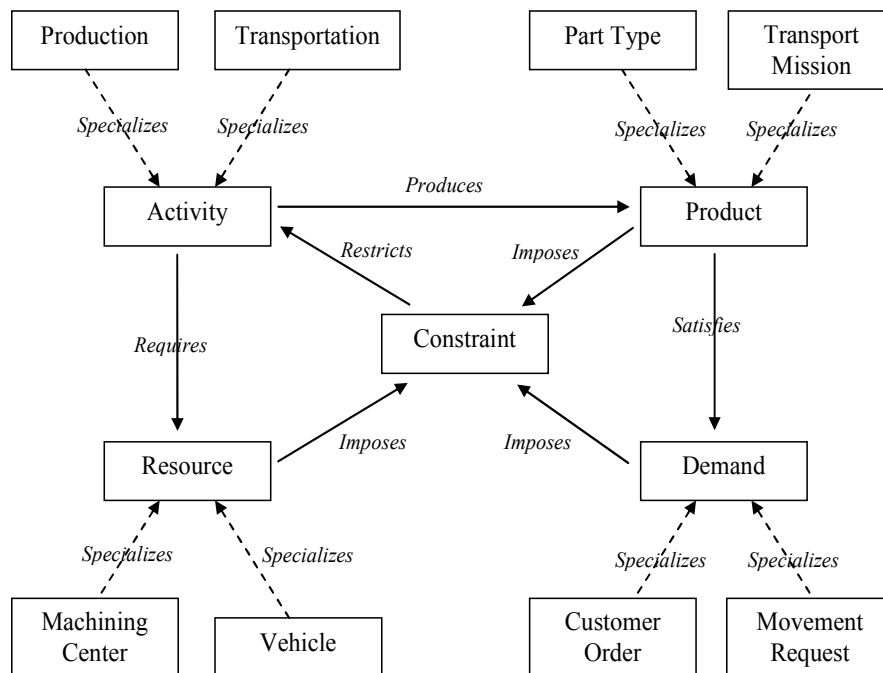


Figure IV.4. Modèle conceptuel de l'ontologie OZONE et ses sous-domaines (d'après (Smith et Becker, 1997))

OZONE s'appuie sur les cinq concepts de base suivants : *Demande* (Demand), *Activité* (Activity), *Ressource* (Resource), *Produit* (Product), et *Contrainte* (Constraint).

Une demande concerne un ou plusieurs produits. La satisfaction des demandes est basée sur l'exécution des activités. Une activité est un processus qui utilise des ressources pour produire et fournir des produits. L'utilisation des ressources et l'exécution des activités sont gérées par un ensemble de contraintes. Dans le contexte d'OZONE, la planification d'un projet est définie comme un processus de synchronisation de l'utilisation des ressources par les activités pour satisfaire les demandes dans le temps. Chaque concept de base peut être spécialisé par des concepts correspondant aux sous-domaines du domaine du concept de base comme le montre la figure IV.

Bien que généralisant les concepts de planification, cette ontologie nécessite des améliorations afin de prendre en compte les particularités de toutes les ontologies manipulées par les applications de planification. Aussi, les modifications effectuées concernent l'ajout de nouvelles classes avec leurs propriétés. L'ajout de la contrainte de coût à la classe *Constraint* non présente dans OZONE permet de prendre en compte les coûts de réalisation des activités définis dans certaines ontologies (Archimede *et al.*, 2003). Pour cela, nous avons ajouté à la classe *Constraint* une sous-classe *Cost_Constraint*, deux propriétés de données (*Cost_Value* et *Cost_Currency*) et une propriété d'objet (*hasCost_Constraint*) pour les concepts *Demand* et *Activity*.

Par ailleurs, afin de mieux structurer le résultat de la planification d'un projet, nous avons complété cette ontologie par les concepts *Planned_Activity*, *Planned_Demand*, et *Activity_Scheduling_Plan*. Les relations entre les classes ajoutées et les classes existantes sont présentées sur la figure IV.5 qui ne présente que la partie concernée par ces modifications.

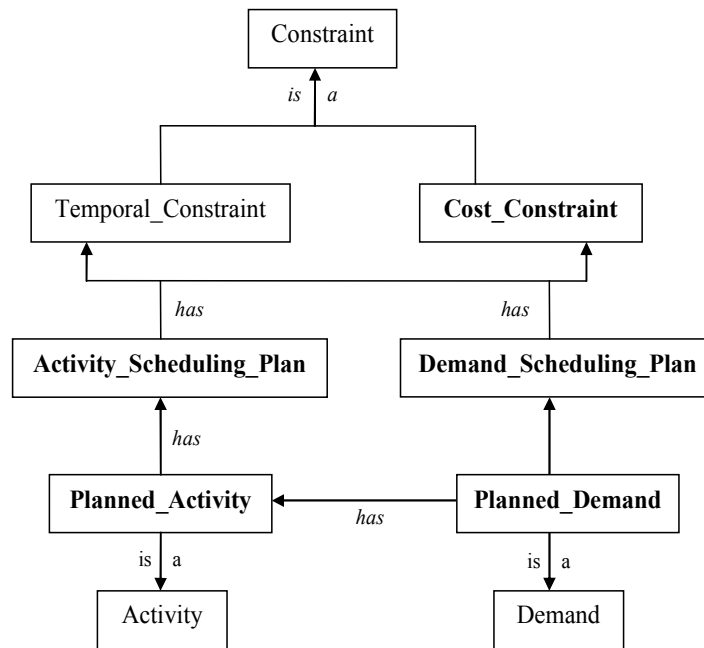


Figure IV.5. Classes ajoutées dans OntoBase et leurs relations

- **Planned_Demand** concerne une demande déjà établie par un client (*hasDemand*). Elle possède une **Cost_Constraint** (*Planned_Demand_Cost*) et des contraintes temporelles (*Start_Date*, *End_Date*), et concerne une ou plusieurs activités planifiées (*hasPlanned_Activities*). En d'autres termes, une **Planned_Demand** comprend l'Ordre de Fabrication (OF) planifié, les contraintes temporelles proposées pour cet OF, le coût, et les activités planifiées.
- **Planned_Activity** concerne une activité existante (*hasActivity*) et possède un ou plusieurs plans d'ordonnancement (*hasActivity_Scheduling_Plans*) ainsi qu'une **Cost_Constraint** (*Planned_Activity_Cost*). Elle contient les détails de planification de chaque activité planifiée (le domaine, le coût, etc.). Du fait qu'une activité planifiée peut être interruptible (le temps de sa réalisation n'est pas continu), la propriété *hasActivity_Scheduling_Plans* de la classe **Planned_Activity** permet de grouper les contraintes temporelles proposées pour l'activité planifiée (*StartTime* et *Duration*). Ces plans d'ordonnancement sont définis dans une nouvelle classe ajoutée aussi dans OntoBase qui est **Activity_Scheduling_Plan** contenant le début et la durée de chaque séquence de réalisation de l'activité planifiée.
- **Activity_Scheduling_Plan** possède des contraintes temporelles concernant le début de l'activité planifiée (*Start_Time*), sa fin (*End_Time*) et sa durée (*Duration*). Une

instance de cette classe peut contenir plusieurs activités si ces activités ont le même début (Start_Time), la même fin (End_Time) et la même durée (Duration).

Tout résultat de planification, pour tout sous ensemble d'activités d'un projet, doit être décrit selon des concepts d'OntoBase par le producteur correspondant avant d'être envoyé à l'agent ambassadeur du client, et plus précisément par les classes présentées sur la figure IV.5 ci-dessus ainsi que leurs propriétés. De même, toute demande d'invocation de service de la part d'un agent ambassadeur doit être décrite selon les concepts d'OntoBase, et plus précisément selon les classes Activity, Product, Demand, et Constraint ainsi que leurs sous-classes et leurs propriétés. Les descriptions des invocations des services et celles des résultats d'invocation selon OntoBase sont mieux illustrées et détaillées dans les sections suivantes décrivant les interactions entre les différents acteurs et les échanges d'informations entre eux.

5 Modèle de l'invocation du service basée sur OntoBase

Les producteurs implémentent le même modèle d'invocation et le même modèle de résultat pour leurs services car le service Web d'un producteur doit être invoqué selon l'ontologie globale OntoBase présentée précédemment et doit fournir un résultat décrit également selon cette même ontologie. Le modèle d'invocation du service est présenté sur la figure IV.6.

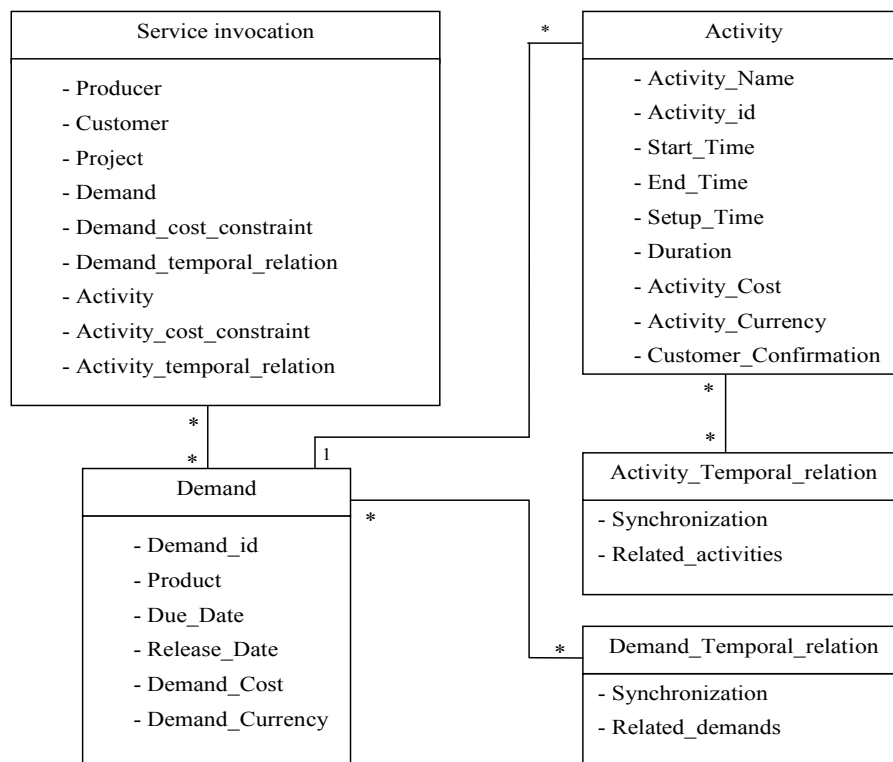


Figure IV.6. Modèle d'invocation du service Web chez les producteurs

L'invocation du service contient l'identifiant du producteur (Producer) fournisseur du service Web, le client (Customer), ses projets (Project), l'ensemble des OF concernés par ces projets (Demand), l'ensemble des contraintes de coûts de réalisation correspondant à ces ordres de fabrications (Demand_Cost_constraint), l'ensemble des contraintes temporelles qui relient les différentes OF d'un projet (Demand_temporal_relation). Aussi, l'invocation du service comprend l'ensemble des activités (Activity) demandées par les différents OF des projets ainsi que leurs contraintes de coût (Activity_cost_constraint) et contraintes temporelles qui relient les activités (Activity_temporal_relation) d'un même projet. Chaque activité comprend le début souhaité (Start_Time), la fin souhaitée (End_Time), la durée souhaitée (Duration), l'activité demandée (Activity_Name) qui peut être une des activités fournies par les producteurs (Assemblage, Teinture, Gravure, Sciage Bois, etc.), ainsi que l'identifiant de l'activité (Activity_id) qui est unique pour la demande à laquelle cette activité appartient.

Chaque demande (OF) comprend un ensemble d'activités (Activity). Chaque activité d'une demande peut posséder une ou plusieurs relations temporelles qui sont définies par le paramètre *Synchronization* afin de déterminer la relation entre cette activité et l'ensemble des activités concernées (Related_activities). Le paramètre de synchronisation peut avoir une des valeurs de l'ensemble {Start_Before, End_Before, Start_After, End_After, Same_Start, Same_End}. De même pour les demandes, chacune peut avoir des relations temporelles (Demand_Temporal_Relation) avec d'autres demandes (Related_demands). Afin de gérer les validations (confirmations) des plans d'ordonnancement proposés par les producteurs pour les projets clients, une contrainte de confirmation (Customer_Confirmation) est rajoutée pour chaque activité. Cette contrainte peut avoir deux valeurs (oui ou non) et permet au client de confirmer au producteur la validation du résultat de l'invocation précédente du service pour que ce dernier prenne en compte cette validation dans son système de planification dans les prochains cycles.

Comme dans l'invocation du service, le résultat de cette invocation est aussi modélisé selon l'ontologie globale OntoBase. Le modèle du résultat d'une invocation du service est donné sur la figure IV.7.

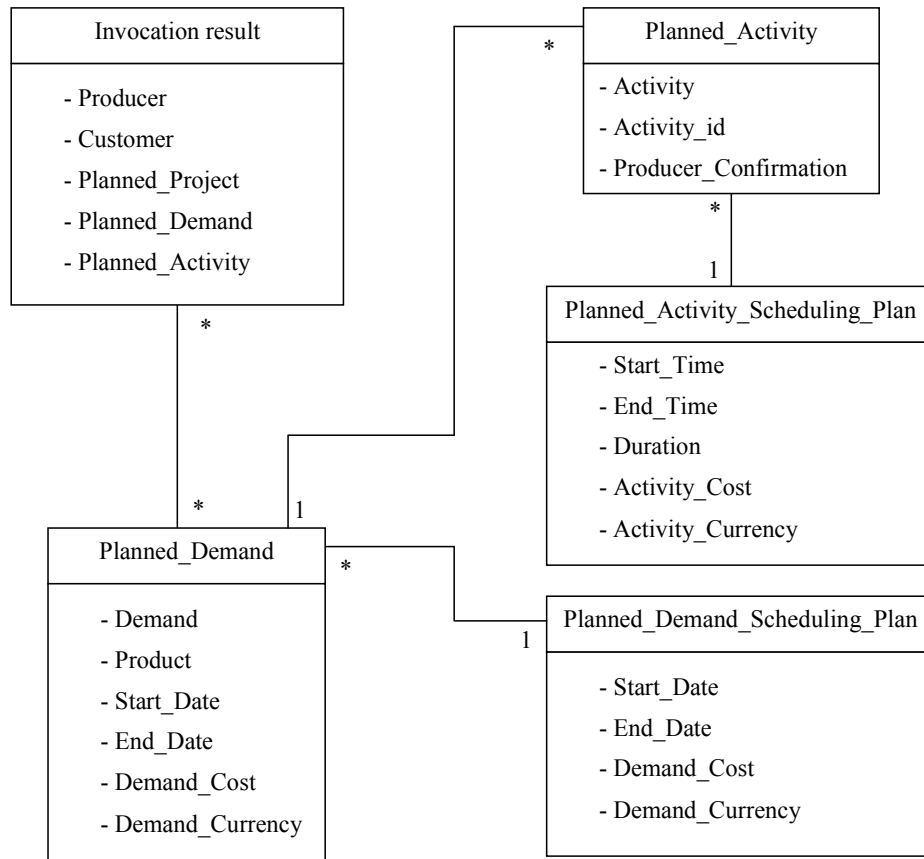


Figure IV.7. Modèle du résultat de l'invocation du service Web chez les producteurs

Le résultat de l'invocation du service comprend le producteur (Producer) qui a généré ce résultat, le client concerné (Customer), ses projets planifiés (Planned_Project), l'ensemble des OF planifiés (Planned_Demand) et leurs contraintes (Planned_Demand_Scheduling_Plan), l'ensemble des activités planifiées (Planned_Activity) et leurs contraintes (Planned_Activity_Scheduling_Plan). Comme le client n'est intéressé que par le début, la fin et la durée totale de l'activité planifiée, un seul plan d'ordonnancement est retourné au client. Cet ordonnancement comprend les contraintes temporelles et le coût de réalisation proposé pour chaque activité planifiée. Comme dans l'invocation du service, un paramètre de confirmation (Producer_Confirmation) est rajouté pour chaque activité planifiée indiquant si le producteur a bien reçu du client la validation du résultat de l'invocation précédente.

Concernant la traduction des informations entre les ontologies (locales et OntoBase), le service de planification du producteur interagit avec deux bases de données (BDP_OntoBase et BDP_TAPAS) basées sur les deux ontologies (respectivement OntoBase et TAPASOntology). La base de données BDP_OntoBase (respectivement BDP_TAPAS) regroupe les informations concernant les demandes d'invocation du service et les résultats des invocations selon l'ontologie globale OntoBase (respectivement l'ontologie locale de

TAPAS). L'agent ATP du producteur a été implémenté sous forme d'une classe *Translate* qui possède deux méthodes *TranslateFromTAPAStoOntoBase* et *TranslateFromOntoBaseToTAPAS* qui effectuent des opérations de lecture et d'écriture dans les bases de données BDP_OntoBase et BDP_TAPAS respectivement. Les différentes interactions entre le service Web et cet agent sont détaillées dans les sections ci-après.

Du côté client, l'agent de traduction est implémenté par la classe *Translate* contenant deux méthodes : *TranslateFromR@msestoOntoBase* pour traduire le projet décrit selon l'ontologie de R@mse en projet décrit selon l'ontologie *OntoBase* afin d'invoquer les services de planification concernés, et *TranslatefromOntoBasetoR@mse* pour traduire le résultat de l'invocation de service qui est un plan d'ordonnancement décrit selon l'ontologie *OntoBase* en résultat décrit selon l'ontologie de R@mse, afin d'être bien interprété par les agents clients.

6 Interactions entre les partenaires et planification des projets

Afin d'élaborer un plan d'ordonnancement multi-site pour la réalisation de son projet, le client interagit avec les producteurs en invoquant les services de planification correspondants. La structure SCEP-SOA du client et des producteurs impliqués dans ces interactions est représentée sur la figure IV.8.

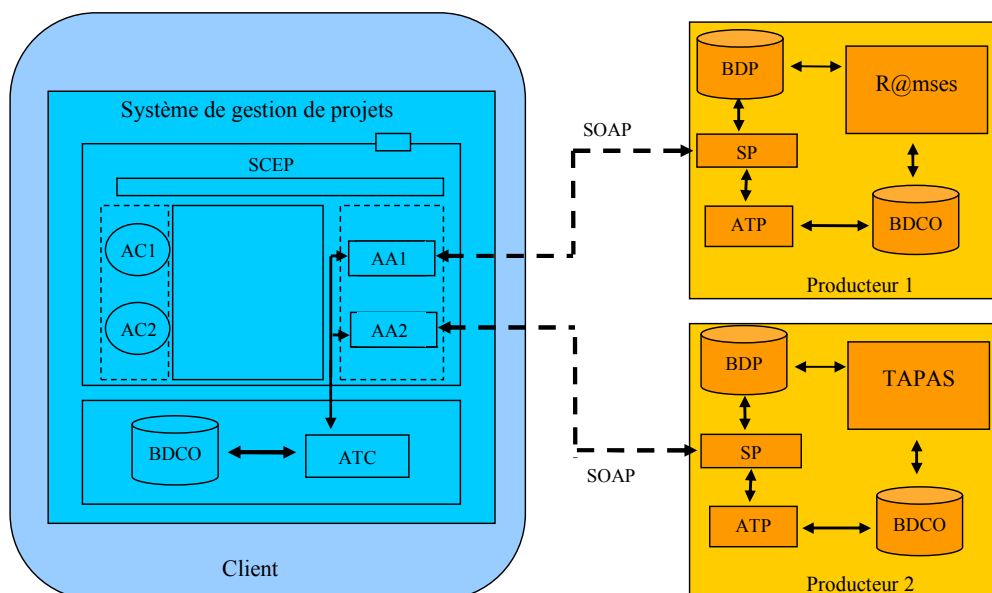


Figure IV.8. Acteurs SCEP-SOA de l'exemple étudié et leurs composants

Pour gérer la réalisation de ses projets, le client implémente dans son modèle SCEP deux agents clients (AC1 et AC2) correspondant respectivement aux deux projets 1 et 2. Deux

agents ambassadeurs AA1 et AA2 sont également implémentés. Ils sont responsables respectivement des interactions avec le producteur 1 et le producteur 2.

La BDP, base de données contenant les projets clients et les résultats de planification de ces projets, ainsi que le système TAPAS chez le producteur 2, manipulent des concepts modélisés par l'ontologie de planification du producteur (TAPASontology).

Chez le client, les agents clients et ambassadeurs manipulent des concepts modélisés par l'ontologie de planification de R@mses (R@msesOntology). Le producteur 1 utilise le même système de planification que le client. Aussi, nous nous intéressons plutôt aux interactions avec le producteur 2 afin de montrer l'interopérabilité, au niveau sémantique, entre les ontologies différentes manipulées par les deux systèmes R@mses et TAPAS.

Afin de rendre son système de planification interopérable, chaque producteur implémente la fonctionnalité de son système dans un service invocable à distance par les clients désirant réaliser des activités fournies par ce producteur. Pour cela, nous avons utilisé les services Web en c# sous Visual Studio.net. Un service Web est une application accessible à partir du Web (Curbera *et al.*, 2001), conçue pour traiter l'interopérabilité de machine-à-machine sur un réseau (Hu, 2003). Il utilise les protocoles Internet pour communiquer et utilise un langage standard pour décrire son interface. Les services web peuvent être définis comme étant une technique permettant à des applications de dialoguer via Internet, par l'échange de messages fondé sur des standards, et ceci indépendamment des plates-formes et des langages sur lesquelles elles reposent.

Nous considérons que chaque producteur implémente un seul service de planification (SP) qui interagit avec le système de planification du producteur afin de planifier les activités fournies par ce dernier. Notons que l'invocation d'un service par le client peut se limiter à une seule activité parmi celles fournies par ce producteur.

L'objectif de cet exemple est de montrer l'interopérabilité entre deux systèmes de planification hétérogènes qui sont R@mses et TAPAS. Nous représentons dans ce qui suit le processus d'un cycle de planification comprenant les étapes décrites dans le chapitre 2 intégrant celles de la deuxième stratégie d'interopérabilité sémantique décrite dans le chapitre 3. Dans ce processus, nous nous intéressons notamment à l'échange d'informations entre les acteurs disposant de systèmes hétérogènes, *i.e.* le client et le producteur 2.

6.1 Récupération des données des projets par les agents ambassadeurs

Après la sélection des producteurs et la création des agents ambassadeurs, ces derniers sont chargés de la planification des projets du client. Chaque agent ambassadeur récupère de l'environnement de SCEP les données nécessaires pour l'invocation du service fourni par le producteur qu'il représente. Dans l'environnement de SCEP, une planification à capacité infinie est publiée par les agents clients pour les projets qu'ils représentent. Cette planification correspond parfaitement aux contraintes souhaitées par le client pour ses projets. Un exemple de la planification des deux projets du client est donné sur la figure IV.9.

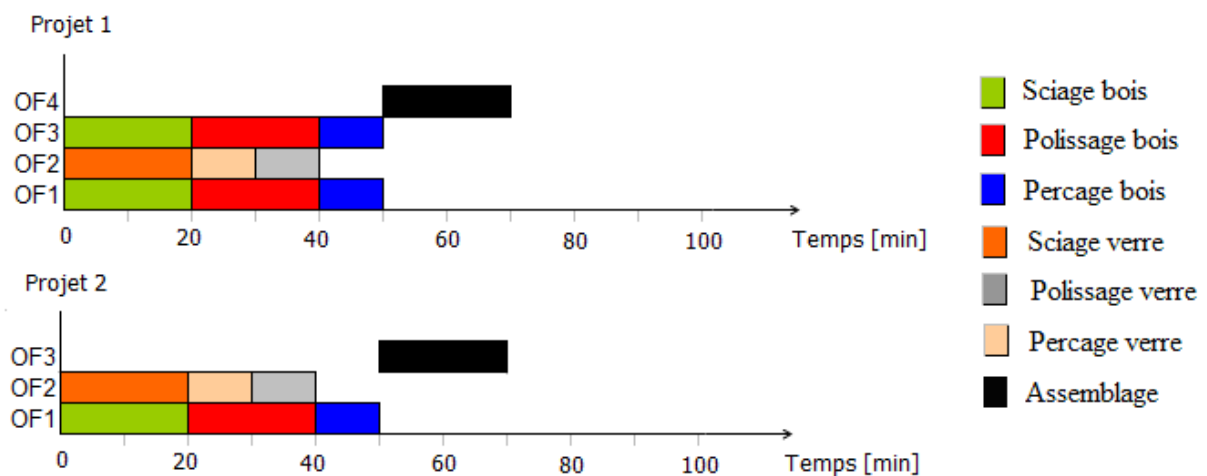


Figure IV.9. Planification souhaitée pour le projet 1 chez le client

Rappelons que le seul lien temporel entre les OF correspond à la réalisation de l'OF4 (respectivement l'OF3) après les OF 1, 2, et 3 (respectivement les OF 1 et 2) dans le projet 1 (respectivement le projet 2).

Les informations échangées entre le client et les différents producteurs concernent les contraintes souhaitées pour la réalisation des projets. Ces contraintes sont modélisées dans l'ontologie de R@mses par les classes MO, Routing, Operation, Operation_Group et leurs propriétés (d'objets ou de type de données). Il s'ajoute à ces contraintes le coût de chaque opération, représenté par la propriété de type de données CostCompetence de la classe Competence de l'ontologie de R@mses.

La figure IV.10 montre une partie de la description du projet 1 définie en format XML selon l'ontologie R@msesOntology.

```

- <Project_Description ID="Project1">
  <Producer>Producteur2</Producer>
- <MO ID="OF1">
  <MO_Due_Date>01/01/2010 00:00:00</MO_Due_Date>
  <MO_Release_Date>01/01/2010 00:50:00</MO_Release_Date>
  <MO_Cost>20$</MO_Cost>
  <MO_Quantity>1</MO_Quantity>
  <Routing>Gamme_pieds</Routing>
  <PreviousMO />
  <NextMO>OF4</NextMO>
- <MO_OperationGroups>
- <MO_OperationGroup ID="grop1">
  <Type>Sequential</Type>
- <Operations>
- <Operation ID="Sciage_pieds">
  <Activity>SciageBois</Activity>
  <Start_Time>01/01/2010 00:00:00</Start_Time>
  <End_Time>01/01/2010 00:20:00</End_Time>
  <Operating_Time>00:20:00</Operating_Time>
  <Setup_Time>00:00:00</Setup_Time>
  <Operation_Cost>10$</Operation_Cost>
  <Customer_Confirmation>No</Customer_Confirmation>
  <PreviousOperation />
  <NextOperation>Percage_pieds</NextOperation>
</Operation>
- <Operation ID="Percage_pieds">
  <Activity>PercageBois</Activity>
  <Start_Time>01/01/2010 00:40:00</Start_Time>
  <End_Time>01/01/2010 00:50:00</End_Time>
  <Operating_Time>00:10:00</Operating_Time>
  <Setup_Time>00:00:00</Setup_Time>
  <Operation_Cost>10$</Operation_Cost>
  <Customer_Confirmation>No</Customer_Confirmation>
  <PreviousOperation>Sciage_pieds</PreviousOperation>
  <NextOperation />
</Operation>
</Operations>
</MO_OperationGroup>
</MO_OperationGroups>
</MO>

```

Figure IV.10. Partie du fichier XML concernant la description du projet 1 décrit selon l'ontologie de R@mses

Ce fichier représente la description de l'OF1 du projet 1. Les balises de ce fichier sont décrites selon les concepts de l'ontologie de R@mses. L'OF1 possède un seul groupe d'opérations séquentielles (grop1) regroupant les opérations de sciage et de polissage des pieds demandant les activités SciageBois et PolissageBois, et doit être réalisé avant l'OF4 correspondant à l'assemblage des éléments d'une chaise. Les relations temporelles entre les opérations du même groupe sont définies par les balises PreviousOperation et NextOperation. Dans le cas

d'une première invocation du service ou en d'autres termes, dans le premier cycle de planification du projet, le paramètre *Customer_Confirmation* de chaque opération est initialisé à Non (No) indiquant au producteur qu'il ne s'agit pas d'une confirmation d'une invocation précédente.

Après la définition d'un projet, chaque agent ambassadeur récupère les informations qui le concerne afin d'invoquer le service de planification chez le producteur correspondant. Par exemple, l'agent ambassadeur AA2, responsable de l'interaction avec le producteur 2 fournissant les activités de Sciage (bois et verre), de perçage et d'assemblage, récupère la description des activités associées appartenant aux quatre OF du projet 1 et aux trois OF du projet 2.

Une fois les informations récupérées par l'agent ambassadeur, ce dernier demande à l'agent de traduction (ATC) leur traduction en informations décrites selon l'ontologie OntoBase afin de les envoyer au producteur correspondant.

6.2 Traduction et envoi des données aux producteurs

Une fois les données reçues par l'ATC, ce dernier récupère de la base de données des correspondances (BDCO) les correspondances entre l'ontologie de R@mses et l'ontologie OntoBase. Il traduit ensuite les informations au format de l'ontologie OntoBase et les envoie à l'agent ambassadeur.

Dans notre cas d'étude, la traduction des informations entre R@msesOntology et OntoBase est faite par la méthode *TranslatefromR@msestoOntoBase*. Cette méthode traduit les informations décrites selon R@msesOntology et contenues dans le fichier XML de la description du projet en informations décrites selon l'ontologie OntoBase en conservant la sémantique de ces informations. Cette méthode appartient à une classe nommée *Translate* qui implémente le fonctionnement de l'agent de traduction ATC chez le client. Un exemple du fichier XML correspondant à la définition de l'ordre de fabrication des pieds de la chaise (OF1) traduit en OntoBase est présenté sur la figure IV.11.

```

- <Project_Description ID="Project1">
  <Producer>Producteur2</Producer>
- <Demand ID="OF1">
  <Due_Date>01/01/2010 00:00:00</Due_Date>
  <Release_Date>01/01/2010 00:50:00</Release_Date>
  <Cost>20</Cost>
  <Currency>$</Currency>
  <Quantity>1</Quantity>
  <Product>Gamme_pieds</Product>
- <Demand_Temporal_Relation>
- <Synchronization ID="Start_Before">
  - <Related_Demands>
    <Demand>OF4</Demand>
    </Related_Demands>
  </Synchronization>
- <Synchronization ID="End_Before">
  - <Related_Demands>
    <Demand>OF4</Demand>
    </Related_Demands>
  </Synchronization>
</Demand_Temporal_Relation>
- <Activity ID="Sciage_pieds">
  <Activity_Name>SciageBois</Activity_Name>
  <Start_Time>01/01/2010 00:00:00</Start_Time>
  <End_Time>01/01/2010 00:20:00</End_Time>
  <Duration>00:10:00</Duration>
  <Setup_Time>00:00:00</Setup_Time>
  <Cost>10</Cost>
  <Currency>$</Currency>
  <Customer_Confirmation>No</Customer_Confirmation>
- <Activity_Temporal_Relation>
  - <Synchronization ID="Start_Before">
    - <Related_Activities>
      <Activity>Percage_pieds</Activity>
      </Related_Activities>
    </Synchronization>
  </Activity_Temporal_Relation>
</Activity>
</Demand>
  - <Related_Activities>
    <Activity>Percage_pieds</Activity>
    </Related_Activities>
  </Synchronization>
- <Synchronization ID="End_Before">
  - <Related_Activities>
    <Activity>Percage_pieds</Activity>
    </Related_Activities>
  </Synchronization>
</Activity_Temporal_Relation>
</Activity>
- <Activity ID="Percage_pieds">
  <Activity_Name>PercageBois</Activity_Name>
  <Start_Time>01/01/2010 00:40:00</Start_Time>
  <End_Time>01/01/2010 00:50:00</End_Time>
  <Duration>00:10:00</Duration>
  <Setup_Time>00:00:00</Setup_Time>
  <Cost>10</Cost>
  <Currency>$</Currency>
  <Customer_Confirmation>No</Customer_Confirmation>
- <Activity_Temporal_Relation>
  - <Synchronization ID="Start_After">
    - <Related_Activities>
      <Activity>Sciage_pieds</Activity>
      </Related_Activities>
    </Synchronization>
  - <Synchronization ID="End_After">
    - <Related_Activities>
      <Activity>Sciage_pieds</Activity>
      </Related_Activities>
    </Synchronization>
  </Activity_Temporal_Relation>
</Activity>
</Demand>

```

Figure IV.11. Partie du fichier XML concernant la description de l'OF1 du projet 1 décrite selon l'ontologie OntoBase

Par souci de gain d'espace, le fichier XML est présenté sur deux colonnes. La comparaison entre les descriptions de l'OF1 du projet 1 proposées aux figures IV.10 et IV.11 montre les correspondances établies de l'ontologie de R@mses vers l'ontologie OntoBase. Nous remarquons que les mêmes informations saisies selon R@msesOntology sont traduites selon l'ontologie OntoBase afin d'être envoyées au producteur correspondant. Il n'y a pas de perte d'informations après la traduction. Les instances de la classe MO de R@msesOntology sont traduites en instances de la classe Demand dans l'ontologie OntoBase en conservant la valeur des contraintes temporelles, quantitatives et économiques définies dans le projet 1.

Les instances de la classe Routing dans l'ontologie de R@mses sont traduites en instances de la classe Product dans l'ontologie OntoBase en regroupant les activités demandées. La balise Demand_Temporal_Relation indique les liens temporels entre l'OF1 et les autres OF (ici OF4). Le lien séquentiel entre l'OF1 et l'OF4, est traduit par des deux balises Synchronization qui indiquent les types des relations temporelles entre les OF (Start_Before et End_Before). La balise Related_Demands regroupe l'ensemble des demandes reliées à la demande en cours par les relations temporelles définies dans les balises Synchronization.

Les instances de la classe Operation et leurs contraintes dans l'ontologie de R@mses sont traduites en instances de la classe Activity de l'ontologie OntoBase en conservant les contraintes temporelles et économiques. L'ID de la balise Activity dans OntoBase correspond à l'identifiant de l'opération (Operation ID) dans R@msesOntology qui doit être unique par OF. La balise Activity_Name d'OntoBase correspond à la balise Activity dans R@msesOntology. Comme pour les activités, la contrainte Activity_Temporal_Relation assure le lien temporel entre les activités pour respecter la séquence demandée dans la définition des groupes d'opérations dans l'ontologie de R@mses.

Les types des liens temporels entre les activités sont représentés par les balises Synchronization. L'activité *Sciage_pieds* est définie avec les valeurs Start_Before et End_Before de la balise Synchronization pour indiquer qu'elle commence et se termine avant l'activité *Polissage_pieds*. Par conséquent, cette dernière doit commencer et se terminer après l'activité *Sciage_pieds* tel que représenté dans la figure IV.11. La balise Related_activites dans une Synchronization regroupe l'ensemble des activités reliées à l'activité concernée par le lien temporel défini dans la balise Synchronization. Pour le type *permutable* d'opérations, il n'y a pas de liens temporels entre les activités c'est-à-dire qu'elles peuvent être planifiées dans n'importe quel ordre, ce qui est illustré dans le fichier XML de la figure 11 par la balise Activity_Temporal_Relation vide. Dans le cas des activités parallèles où les activités en relation doivent avoir le même début et la même fin les valeurs de la balise Synchronization sont Same_Start (même début) et Same_End (même fin).

Après la réception des informations traduites selon OntoBase, le module d'invocation (MI) de l'agent ambassadeur génère une requête SOAP et invoque le service de planification (PS) chez le producteur correspondant.

6.3 Réception, traduction et enregistrement du projet chez les producteurs

En recevant les informations contenues dans la requête d'invocation du service, le service enregistre ces informations dans la base de données BDP_OntoBase et appelle ensuite la méthode *TranslatefromOntoBasetoTAPAS* de la classe Translate qui implémente le fonctionnement de l'agent de traduction (ATP). Les tables de la base de données BDP_OntoBase sont implémentées selon l'ontologie OntoBase et structurées selon le modèle d'invocation du service présenté précédemment. *TranslatefromOntoBasetoTAPAS* traduit les

informations contenues dans l'invocation du service de l'ontologie globale OntoBase au format de l'ontologie locale TAPASontology.

Les correspondances ontologiques pour la traduction des projets clients, depuis OntoBase vers l'ontologie de TAPAS, sont réalisées entre les concepts ou classes Activity, Demand, Activity_Temporal_Relation et Demand_Temporal_Relation issus du modèle d'invocation du service implémenté selon OntoBase et les concepts MO, Routing, et Operation avec ses sous-concepts Phase, Transport et Synchronisation de l'ontologie de TAPAS. Les instances de la classe Demand et ses propriétés (attributs) dans OntoBase sont mappées en instances de la classe MO de l'ontologie de TAPAS en faisant abstraction des contraintes de coût du fait qu'elles ne sont pas considérées dans le modèle de TAPAS. Les instances de la classe Activity et ses propriétés dans OntoBase sont mappées en instances de la classe Operation de l'ontologie de TAPAS, sans tenir compte des contraintes du coût ainsi que des dates de début et de fin des activités qui ne sont pas pris en compte dans l'ontologie de TAPAS. Le temps de préparation (Setup_time) et la durée d'une activité (duration), sont exprimés en secondes dans l'ontologie de TAPAS.

La planification concerne la répartition des activités sur les différentes ressources (machines) chez le producteur. En fait, les ressources sont déjà décrites dans le système TAPAS et plus précisément dans un fichier ressources.res qui contient la description selon l'ontologie de TAPAS des ressources disponibles ainsi que les activités réalisées par ces ressources. Dans notre étude, les ressources concernées sont les Producer, et les Mover qui assurent les opérations de transport intra-site.

Un exemple du fichier ressources.res est donné sur la figure IV.12 qui montre la description des ressources chez le producteur 2 offrant les activités de sciage (bois et verre), de perçage (bois et verre) et d'assemblage.

```

Producer    PercageProducer
Capacity    1
Location    0 0
/*****Task List ( PercageProducer ) *****/
/*****Conflicts List ( PercageProducer ) *****/

Producer    AssemblageProducer
Capacity    1
Location    0 0
/*****Task List ( AssemblageProducer ) *****/
/*****Conflicts List ( AssemblageProducer ) *****/

Producer    SciageProducer
Capacity    1
Location    0 0
/*****Task List ( SciageProducer ) *****/
/*****Conflicts List ( SciageProducer ) *****/

Mover       Robot1
Capacity    1
Path
    AssemblageProducer 0:0:0
    PercageProducer 0:1:0
    AssemblageProducer 0:1:0
End
/*****Task List ( Robot1 ) *****/
/*****Conflicts List ( Robot1 ) *****/

Mover       Robot2
Capacity    1
Path
    AssemblageProducer 0:0:0
    SciageProducer 0:1:0
    AssemblageProducer 0:1:0
End
/*****Task List ( Robot2 ) *****/
/*****Conflicts List ( Robot2 ) *****/

Mover       Robot3
Capacity    1
Path
    SciageProducer 0:0:0
    PercageProducer 0:1:0
    SciageProducer 0:1:0
End
/*****Task List ( Robot3 ) *****/
/*****Conflicts List ( Robot3 ) *****/

```

Figure IV.12. Description des ressources selon l'ontologie de TAPAS

Les trois machines du producteur sont notées par PercageProducer, AssemblageProducer et SciageProducer. Les activités de PerçageBois et PerçageVerre sont réalisées par PercageProducer. Les activités de SciageBois et SciageVerre sont réalisées par SciageProducer. L'activité d'assemblage est réalisée par AssemblageProducer. Trois robots assurent le transport intra-site entre les machines chez ce producteur. La durée d'une activité de transport intra-site est fixée à 1 minute et les machines ont la même capacité qui est égal à 1 indiquant que la machine traite une seule pièce en même temps. Notons la description des ressources est défini par le producteur indépendamment du contenu des projets client.

Pour prendre en compte les liens temporels entre les demandes (OF), des ressources virtuelles nommées Sémaphore dans l'ontologie de TAPAS sont ajoutées entre les OF afin de synchroniser leur ordonnancement selon la séquence définie dans la propriété *Synchronization* de la classe Demand_Temporal_Relation de l'ontologie OntoBase. Cela se fait lors de la description des gammes (Routing) demandées par les OF du projet client. Un exemple est illustré sur la figure IV.13.

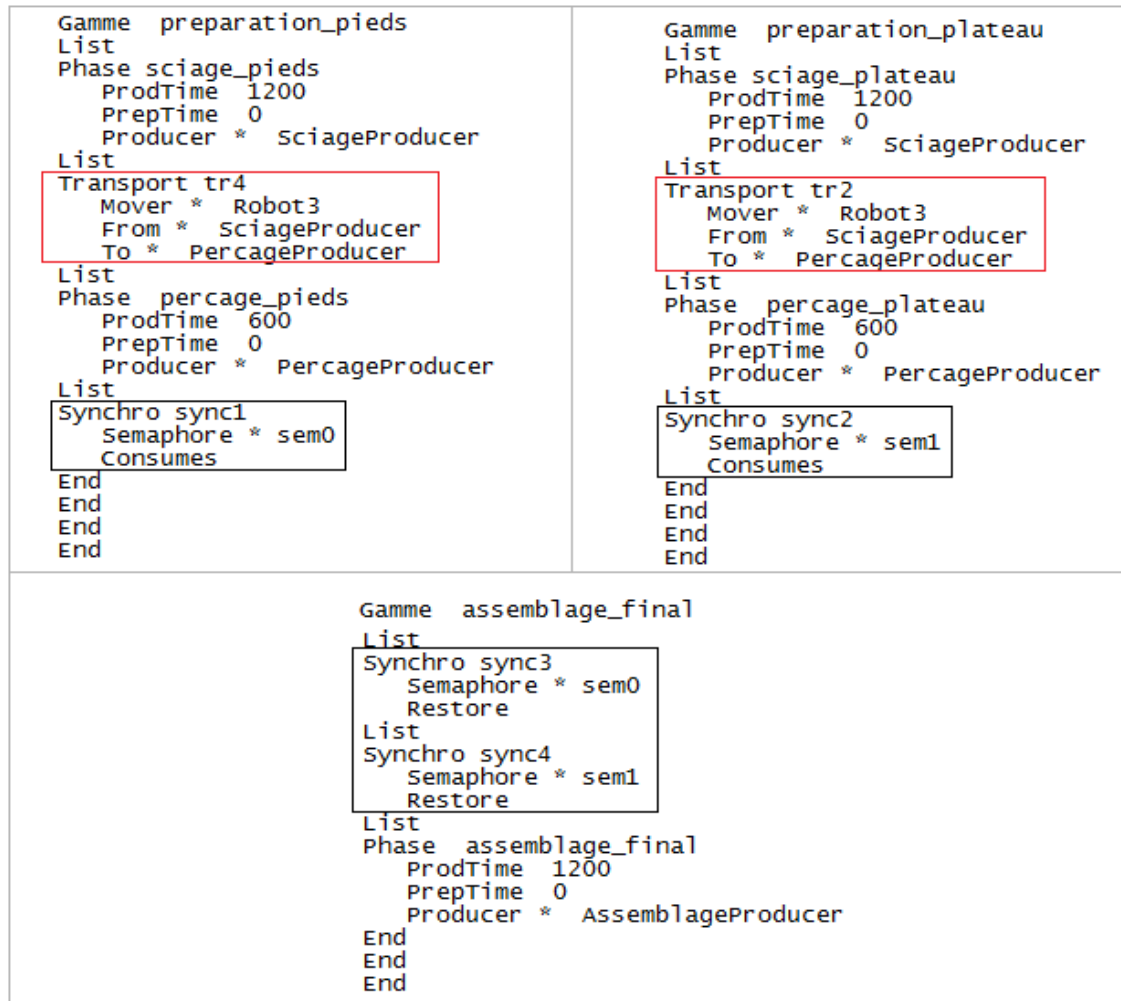


Figure IV.13. Description selon l'ontologie de TAPAS des gammes des OF 1, 2 et 3 du projet 2 chez le producteur 2

Les OF 1, 2 et 3 du projet 2 concernant la fabrication d'une table demandent respectivement les gammes préparation_assise, préparation_dossier, et assemblage_final. Dans la définition du projet 2, les OF 1 et 2 doivent être réalisés avant l'OF3 qui est l'assemblage final. Afin de prendre en compte le lien temporel entre ces OF, le système TAPAS introduit une ressource virtuelle *Semaphore* qui est consommée (Consumes) par un OF et libérée (Restore) par un autre indiquant la séquentialité entre les OF en considérant qu'une *Semaphore* ne peut pas

être libérée avant qu'elle soit consommée. L'utilisation de cette ressource virtuelle (Semaphore) est réalisée par une opération de synchronisation (Synchro). Durant la phase d'ordonnancement, le système TAPAS prend en compte le lien temporel entre ces OF en planifiant l'OF1 et l'OF2 avant l'OF3 comme demandé car la réalisation de l'OF3 est bloqué jusqu'à la réalisation des deux autres OF comme illustré sur la figure IV.13 ci-dessus (les encadrés en noir). Dans l'ontologie de TAPAS, la durée d'une opération est exprimée en secondes par le concept ProdTime.

Concernant les liens temporels entre les activités, le système TAPAS prend en compte deux types : séquentiel et parallèle. Le type permutable de l'ontologie de R@mses n'est pas pris en compte dans l'ontologie de TAPAS.

Notons que dans l'ontologie de TAPAS, le concept de groupe d'opérations n'existe pas. Toutes les activités de l'OF sont groupées sous la même gamme en indiquant *List* devant l'opération qui est représenté par *Phase* dans TAPAS. Si les opérations demandées sont parallèles, l'ensemble de ces opérations (phases) est groupé sous la même liste (List) sans tenir compte du transport intra-site. Cependant, comme le transport intra-site (entre les machines) n'est pas négligeable pour des opérations séquentielles, cela se traduit par l'ajout des opérations de transport (Transport) pour assurer le transport entre les machines comme illustré sur la figure IV.13 ci-dessus (les encadrés en rouge). L'opération de transport est assurée par un robot qui effectue un mouvement entre deux machines (ici Robot3 qui se déplace de SciageProducer à PercageProducer).

La traduction du projet chez ce producteur montre la prise en compte de la séquentialité entre les activités demandées par le client, ce qui illustre la conservation de la sémantique des données échangées.

Une fois les projets traduits, ils sont stockés dans la base de données BDP_TAPAS afin d'être planifiés par le système TAPAS.

6.4 Planification et stockage du résultat

Pour planifier le projet client chez le producteur 3 par exemple, le système de planification de ce producteur récupère de la base de données BDP_TAPAS les informations sur le projet client décrit selon l'ontologie de TAPAS et effectue la planification. Après le lancement du

système, un plan d'ordonnancement est établi sous la forme d'un diagramme de Gant. Le résultat de planification des projets chez le producteur 2 est présenté sur la figure IV.14.

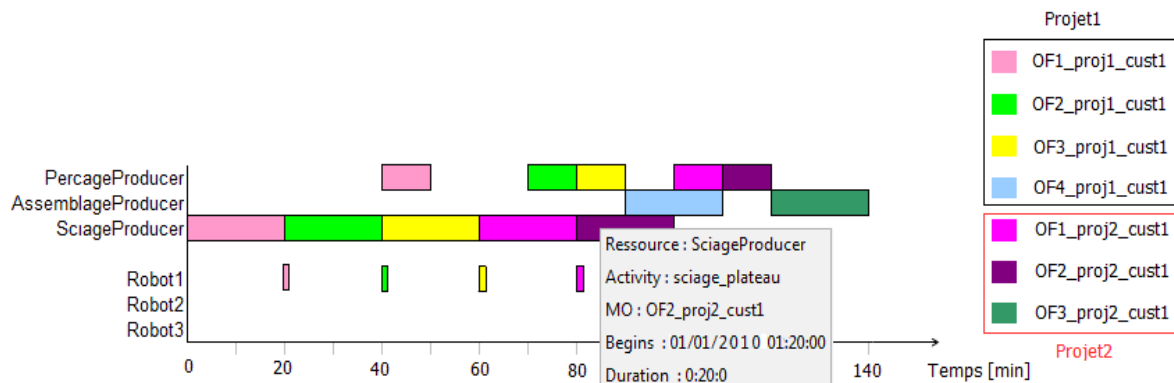


Figure IV.14. Exemple d'un résultat d'ordonnancement chez le producteur 2

Rappelons qu'un projet client chez un producteur comprend seulement les activités du projet réalisables par ce producteur. Pour chaque tâche planifiée, le diagramme indique la ressource *i.e.* SciageProducer, l'activité associée *i.e.* sciage_plateau, l'ordre de fabrication correspondant *i.e.* OF2_proj1_cust1, le début de la tâche et sa durée. Comme les OF peuvent avoir le même nom pour différents clients ou même dans différents projets pour le même client, le producteur nomme les OF en rajoutant l'identifiant du projet *i.e.* proj1, suivi de l'identifiant du client *i.e.* cust1. Nous pouvons constater le respect de la liaison temporelle entre les OFs. Après la planification des projets, le résultat de chaque projet est stocké dans la base de données BDP_TAPAS. Ce résultat est décrit selon les concepts de l'ontologie de TAPAS.

6.5 Traduction et renvoi du résultat au client

Après le dépôt du plan d'ordonnancement résultant de la planification du projet client dans la BDP_TAPAS, le service de planification du producteur doit traduire le résultat décrit selon l'ontologie de TAPAS en résultat décrit selon l'ontologie globale OntoBase avant de l'envoyer à l'agent ambassadeur correspondant. Le service de planification du producteur récupère de la BDP_TAPAS les informations concernant le résultat de la planification. Ces informations étant décrites selon l'ontologie de TAPAS, le service de planification demande à l'agent ATP de traduire les données concernant le résultat de la planification décrites selon l'ontologie de TAPAS en données décrites selon l'ontologie OntoBase. Cet agent se base sur les correspondances ontologiques pour traduire les informations et renvoie le résultat traduit

au service de planification. Un exemple de résultat de planification avant et après la traduction chez le producteur 2 est illustré sur la figure IV.15.

<pre> - <Project_Planning ID="Project1"> <Customer>cust1</Customer> - <MO ID="OF1_project1_cust1"> <Start>01/01/2010 00:00:00</Start> <End>01/01/2010 00:50:00</End> - <Activity ID="sciage_pieds"> <Resource>SciageProducer</Resource> <Begins>01/01/2010 00:00:00</Begins> <Duration>0:20:0</Duration> </Activity> - <Activity ID="percege_pieds"> <Resource>PercegeProducer</Resource> <Begins>01/01/2010 00:40:00</Begins> <Duration>0:10:0</Duration> </Activity> </MO> - <MO ID="OF3_project1_cust1"> <Start>01/01/2010 01:40:00</Start> <End>01/01/2010 02:00:00</End> - <Activity ID="percege_assise"> <Resource>PercegeProducer</Resource> <Begins>01/01/2010 01:40:00</Begins> <Duration>0:20:0</Duration> </Activity> </MO> </pre>	<pre> - <Project_Planning ID="Project1"> <Customer>cust1</Customer> - <Planned_Demand ID="OF1"> - <Planned_Demand_Scheduling_Plan> <Start_Date>01/01/2010 00:00:00</Start_Date> <End_Date>01/01/2010 00:50:00</End_Date> <Demand_Cost>20</Demand_Cost> <Demand_Currency>\$</Demand_Currency> </Planned_Demand_Scheduling_Plan> - <Planned_Activity ID="sciage_pieds"> - <Planned_Activity_Scheduling_Plan> <Activity_Name>SciageBois</Activity_Name> <Start_Time>01/01/2010 00:00:00</Start_Time> <End_Time>01/01/2010 00:20:00</End_Time> <Duration>00:20:00</Duration> <Activity_Cost>10</Activity_Cost> <Activity_Currency>\$</Activity_Currency> <Producer_Confirmation>No</Producer_Confirmation> </Planned_Activity_Scheduling_Plan> </Planned_Activity> - <Planned_Activity ID="percege_pieds"> - <Planned_Activity_Scheduling_Plan> <Activity_Name>PercegeBois</Activity_Name> <Start_Time>01/01/2010 00:40:00</Start_Time> <End_Time>01/01/2010 00:50:00</End_Time> <Duration>00:10:00</Duration> <Activity_Cost>10</Activity_Cost> <Activity_Currency>\$</Activity_Currency> <Producer_Confirmation>No</Producer_Confirmation> </Planned_Activity_Scheduling_Plan> </Planned_Activity> </Planned_Demand> </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figure IV.15. Résultat de planification décrit selon les deux ontologies TAPASOntology et OntoBase chez le producteur 2

La définition du résultat de la planification selon l'ontologie de TAPAS est donnée sur la partie gauche de la figure IV.15. Ce résultat est établi par le système TAPAS du producteur 2. Les mêmes informations sont décrites selon l'ontologie OntoBase en ajoutant certains concepts comme illustré sur la partie droite de la figure IV.15. Pour la limite d'espace, seul le résultat de planification de l'OF1 du projet 1 est représenté en OntoBase sur la partie droite de la figure.

Dans le résultat décrit en OntoBase, les contraintes de coût sont récupérées de la description du projet stockée dans la BDP_OntoBase et rajoutées au résultat. Ce concept (coût) n'étant pas pris en compte par le système TAPAS mais nécessaire pour le système R@mses, ce qui montre la conservation de la sémantique des informations échangées.

Pour chaque activité dans OntoBase, le nom (Activity_Name), la date de début (Start_Time), la date de fin (End_Time) et la durée (Duration) sont représentées. Ces contraintes

correspondent respectivement à la date de début de la première tâche de planification de l'activité, à la date de fin de la dernière tâche de planification de l'activité et à sa durée totale. Le système TAPAS peut planifier une activité en plusieurs tâches du fait que les activités sont considérées interruptibles. Chaque tâche a une date de début d et une durée dr . La date de fin de chaque tâche d'une activité est calculée en ajoutant la durée à la date de début donc la date de fin $f = d + dr$ pour une activité planifiée. Si t_1 est la première tâche et t_n est la dernière tâche d'une activité planifiée, alors la date de début (Start_Time), la date de fin (End_Time), et la durée totale (Duration) sont données par les formules suivantes :

- Start_time = $d(t_1)$
- End_Time = $f(t_n) = d(t_n) + dr(t_n)$
- Duration = $\sum_{i=1}^n dr(t_i)$

En recevant le résultat traduit en OntoBase, le service de planification génère une réponse SOAP contenant le résultat traduit et le renvoie à l'agent ambassadeur correspondant comme résultat de l'invocation du service.

6.6 Traduction et dépôt du résultat dans l'environnement SCEP du client

En recevant le résultat de la planification du producteur décrit selon l'ontologie OntoBase, l'agent ambassadeur du client, ou plus précisément le module d'invocation MI de cet agent, demande à l'agent de traduction (ATC) de traduire selon $R@msesOntology$ les informations contenues dans ce résultat avant de les déposer dans l'environnement de SCEP afin qu'elles soient bien interprétées par les agents clients correspondants.

En se basant sur les correspondances ontologiques entre OntoBase et $R@msesOntology$ contenues dans la BDCO, l'ATC traduit les informations contenues dans le résultat d'invocation correspondant à la planification du projet client. Dans notre exemple, cette traduction est réalisée par la méthode *TranslatefromOntoBasetoR@mses* de la classe *Translate* qui implémente l'agent ATC du client. Un exemple de résultat pour le projet 1 reçu du producteur 2 et décrit selon $R@msesOntology$ est illustré sur la figure IV.16.

```

- <Project_Planning ID="Project1">
  <Producer>producteur2</Producer>
- <MO ID="OF1">
  <Start>01/01/2010 00:00:00</Start>
  <End>01/01/2010 00:50:00</End>
  <Cost>20$</Cost>
- <Operation ID="sciage_pieds" Activity="SciageBois">
  <Start>01/01/2010 00:00:00</Start>
  <End>01/01/2010 00:20:00</End>
  <Duration>00:20:00</Duration>
  <Cost>10$</Cost>
  <Producer_Confirmation>No</Producer_Confirmation>
</Operation>
- <Operation ID="perçage_pieds" Activity="PerçageBois">
  <Start>01/01/2010 00:40:00</Start>
  <End>01/01/2010 00:50:00</End>
  <Duration>00:10:00</Duration>
  <Cost>10$</Cost>
  <Producer_Confirmation>No</Producer_Confirmation>
</Operation>
</MO>

```

Figure IV.16. Fichier XML du résultat de planification reçu du producteur 2 et traduit en R@msesOntology

Le résultat présenté concerne la planification de l'OF1 du projet 1 pour les activités de sciage et perçage fournies par le producteur 2. Le contenu du résultat décrit en OntoBase est traduit en résultat décrit selon l'ontologie du client R@msesOntology. Les mêmes informations sont décrites mais structurées selon l'ontologie de R@mses.

Une fois le résultat traduit, le module d'invocation MI de l'agent ambassadeur transmet le fichier XML du résultat traduit au module de récupération MR de cet agent. Ensuite, le MR dépose les informations décrites selon l'ontologie de R@mses dans l'environnement SCEP afin d'être découvert et interprété d'une manière compréhensible par l'agent client chargé de la gestion de ce projet.

6.7 Etude et validation du résultat proposé

Une fois les résultats traduits en R@msesOntology publiés par les agents ambassadeurs dans l'environnement SCEP, l'agent client responsable de la gestion du projet client récupère les plans d'ordonnancement établis pour ce projet pour les étudier. Les plans d'ordonnancement pour le projet 1 reçus des producteurs 1 et 2 sont représentés sur les figures 4.17 et 4.18 respectivement.

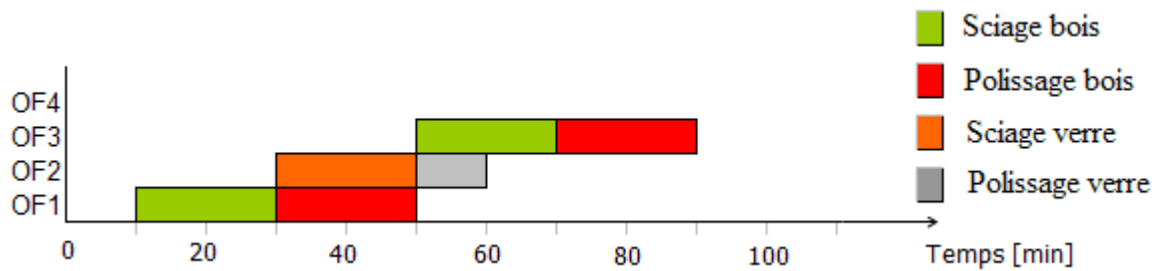


Figure IV.17. Résultat de planification du projet 1 reçu du producteur 1

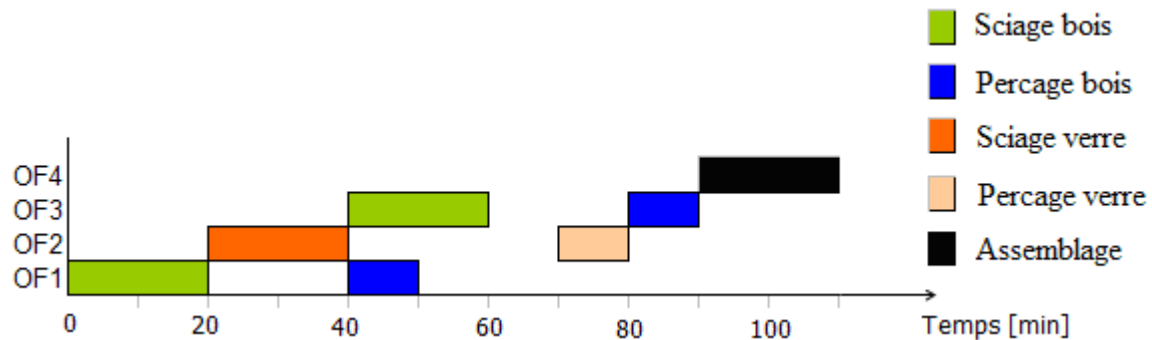


Figure IV.18. Résultat de planification du projet 1 reçu du producteur 2

Les activités de sciage (bois et verre) sont concernées par ces deux plans. Cela provient du fait que ces activités sont fournies par les deux producteurs. Nous remarquons que dans le résultat reçu du producteur 2 où tous les OF sont concernés, les liens temporels entre les OF sont bien respectés où l'OF4 est bien planifié après les autres OF. Comme il n'y a pas de liens temporels entre les autres OF, des chevauchements sont possibles entre les OF 1, 2 et 3. Dans le résultat reçu du producteur 1, la dernière activité de l'OF3, *i.e.* polissage bois, se termine avant le début de l'activité d'assemblage de l'OF4 dans le plan proposé par le producteur 2.

Dans l'étude du résultat, l'agent client construit un plan global du projet client comprenant toutes les activités demandées par ce projet. En fait, ce plan global est construit en se basant sur les ordonnancements publiés par les différents agents ambassadeurs impliqués dans le projet client. En étudiant l'ordonnancement du projet, l'agent client peut valider une ou plusieurs activités planifiées ou même la totalité du plan du projet. Pour les résultats représentés sur les figures 4.17 et 4.18, les deux meilleurs choix de validation des résultats par le client sont les suivants :

- Le client peut valider la totalité du résultat reçu du producteur 2 pour le projet 1. Dans ce cas, il demande la re-planification de l'activité de polissage bois pour les OF 1 et 3 chez le producteur 1 qui fournit cette activité. Cette demande doit prendre en compte le résultat validé du producteur 2. En d'autres termes, la demande de re-planification

contient des nouvelles contraintes pour l'activité de polissage bois afin qu'elle soit cohérente avec le résultat validé, *i.e.* entre 20 et 40 pour l'OF1 et entre 60 et 80 pour l'OF3.

- Le client peut valider la totalité du résultat reçu du producteur 1. Dans ce cas, il demande la re-planification des activités de perçage bois pour l'OF1 et l'OF3 et d'assemblage pour l'OF4. Cette demande doit prendre en compte le résultat validé du producteur 1. En d'autres termes, la demande de re-planification contient des nouvelles contraintes pour l'activité de perçage bois pour les OF1 et 3 et de l'activité d'assemblage pour l'OF4 afin qu'elles soient cohérentes avec le résultat validé, *i.e.* après 50 (respectivement 90) pour l'activité de perçage bois dans l'OF1 (respectivement l'OF3).

Afin de communiquer le résultat de la validation aux producteurs, une nouvelle invocation de service est nécessaire par les agents ambassadeurs. Dans cette nouvelle invocation, l'agent ambassadeur initialise le paramètre `Customer_Confirmation` en "Yes" pour chaque activité planifiée et validée par l'agent client et par "No" pour celles non validées. Si dans cette nouvelle invocation il y a des activités validées et que la réponse du client arrive chez le producteur après la suppression du résultat chez ce producteur, le client recevra un plan d'ordonnancement avec des valeurs "No" pour le paramètre `Producer_Confirmation` pour les activités planifiées. La réception d'un tel résultat par le client lui indique que sa validation pour le résultat précédent n'a pas été prise en compte par le producteur.

Le projet est considéré planifié si pour toutes les activités du projet, le client a validé des résultats et que ce client a reçu des réponses des producteurs correspondants confirmant cette validation. Cela correspondant à l'obtention des valeurs "Yes" pour le paramètre `Producer_Confirmation` dans les résultats d'invocation pour toutes les activités planifiées par les producteurs.

Le cas étudié concerne un seul cycle de planification. Le même processus est répété pour tous les autres cycles jusqu'à l'obtention d'un plan d'ordonnancement valide pour chaque projet client. Dans le meilleur des cas, le nombre de cycles est égal à 2 correspondant à l'obtention d'un plan valide pour tous les projets du client dès la première invocation des services correspondants. Le premier cycle correspond à l'invocation des services et à l'obtention d'un premier résultat. Le deuxième cycle correspond à la ré-invocation des services concernés pour

la validation des résultats obtenus, et à l'obtention de la confirmation de ces validations de la part des producteurs concernés.

7 Conclusion

Dans ce chapitre, l'interopérabilité sémantique dans SCEP-SOA a été illustrée grâce à l'application de la stratégie d'interopérabilité sémantique basée sur l'approche hybride d'ontologies présentée dans le chapitre précédent. L'application de cette stratégie a montré la conservation de la sémantique des informations échangées entre les partenaires disposant de systèmes de planification hétérogènes (R@mses et TAPAS). Les concepts ajoutés à l'ontologie OZONE dans OntoBase permettent de mieux structurer la sémantique des concepts de planification échangés.

Le cas d'étude présenté montre la faisabilité de la stratégie sémantique adoptée garantissant l'interopérabilité sémantique dans l'échange d'informations entre les systèmes de planification hétérogènes. Il serait intéressant d'étudier l'application de cette stratégie pour d'autres systèmes de planification ainsi que pour des systèmes dédiés à d'autres fonctionnalités de gestion, *i.e.* conception, commercialisation, etc.

Conclusion générale et perspectives

1 Conclusion générale

Aujourd'hui, avec Internet et les avancées sur les technologies de la communication, les grands groupes ont mis en place des places de marché électroniques afin de gérer les relations avec leurs sous-traitants. Dans ces marchés, les contraintes fortes imposées pénalisent un grand nombre de PME ayant des capacités limitées. Dans le premier chapitre, nous avons étudié le marché orienté-service qui améliore la compétitivité des PME et augmente les opportunités d'intégration de ces dernières dans les marchés de production actuels. Malgré cela, les collaborations entre les partenaires dans un tel marché restent difficiles du fait de la distribution de la gestion des projets qui met en évidence les hétérogénéités existant entre les modèles et les technologies utilisés par les systèmes et les applications des partenaires.

Pour gérer cette distribution, les systèmes multi-agents sont adaptés à ce contexte. Le modèle SCEP proposé et développé au Laboratoire Génie de Production de l'ENI de Tarbes permet une collaboration entre différents agents afin de gérer la réalisation des projets multi-sites. Le modèle SOA offre par ailleurs l'infrastructure d'un marché orienté-service permettant la mise en relation des partenaires grâce au registre de services ainsi qu'une gestion de l'hétérogénéité grâce au concept de service.

Basée sur l'intégration des concepts de ces deux modèles, une architecture SCEP-SOA a été proposée dans le chapitre 2. Elle permet une gestion distribuée des projets de production multi-sites et rend interopérable le modèle SCEP et les différents modèles et systèmes utilisés par les partenaires grâce à la notion de service SOA. La spécification et le fonctionnement de cette architecture mettent en œuvre les concepts du marché orienté-service présenté dans le premier chapitre et permet une coopération et une collaboration entre les différents systèmes de production utilisés par les entreprises partenaires indépendamment de toute spécificité de l'une de ces dernières.

Cependant on identifie un manque d'interopérabilité sémantique, car les partenaires impliqués dans un projet multi-site n'ont pas la même connaissance du domaine et ils utilisent des concepts voisins et des vocabulaires différents. Les ontologies étudiées dans le chapitre 3 sont

actuellement en cours d'application dans la majorité de domaines industriels et commerciaux nécessitant un échange électronique des données entre des systèmes et applications hétérogènes. Ces ontologies sont des outils qui capturent la connaissance d'un domaine chez un partenaire. L'hétérogénéité entre les ontologies est un problème important qui existe dans l'architecture SCEP-SOA du fait que les partenaires ont chacun leur propre ontologie.

Bien que l'approche multi-ontologique, basée sur l'établissement de correspondances directes entre les ontologies locales des partenaires, permette de résoudre le problème d'hétérogénéité entre les ontologies, son application dans le contexte de SCEP-SOA est complexe. Cela provient de la complexité du raisonnement avec les correspondances ontologiques à établir. Cette complexité doit prendre en compte les spécificités des différentes ontologies utilisées par les partenaires. L'approche hybride semble être plus adaptée à notre contexte du fait qu'elle réduise la complexité des correspondances ontologiques et qu'elle assure une convergence sémantique vers une ontologie globale et commune utilisée dans l'échange des informations entre les partenaires.

Le cas d'étude présenté dans le dernier chapitre pour illustrer l'interopérabilité entre des applications de planification hétérogènes montre bien le rôle et l'intérêt de l'ontologie de planification globale "OntoBase" dans l'échange interopérable des informations entre les partenaires. L'application de la stratégie sémantique basée sur l'approche hybride dans SCEP-SOA montre l'adaptation de cette stratégie à notre contexte et illustre la complémentarité entre les SMAs, le modèle SOA et les ontologies dans les collaborations inter-entreprises.

2 Perspectives

Bien que l'exemple présenté dans le dernier chapitre illustre le niveau d'interopérabilité, et notamment l'interopérabilité sémantique, assurée par SCEP-SOA lors des échanges et des interactions entre des applications de planification hétérogènes, nous n'avons pas traité dans l'implémentation réalisée le cas de la découverte et la publication des services. En fait, l'interopérabilité sémantique illustrée concerne l'étape d'invocation des services et les interactions entre le client et ses partenaires en considérant que le client a déjà identifié ses partenaires dans le registre des services. En outre, des modèles d'invocation de service et de réponse d'invocation ont été aussi présentés dans ce dernier chapitre définissant la structure des données encapsulées dans ces interactions.

Cependant, les modèles de publication et de découverte de services n'ont pas été traités. Bien que ces deux fonctions soient considérées elles-mêmes comme des services fournis par le registre, néanmoins, elles sont demandées par des acteurs différents qui ont leurs propres termes et concepts de description et de recherche, ce qui remet en évidence la problématique de l'interopérabilité sémantique à ce niveau. Une piste de recherche dans cet objectif peut être l'introduction de l'ontologie OWL-S (Semantic Markup for Web Services)* ou WSMO (Web Service Modeling Ontology)** par exemple dans la publication et la découverte des services dans le registre. Cette ontologie peut être couplée avec l'ontologie de base du domaine pour garantir l'interopérabilité sémantique entre les applications mises en jeu.

Dans la définition des projets chez le client, nous n'avons pas pris en compte le transport inter-site. La prise en compte de ce transport va exiger des synchronisations dans l'invocation des services qui devraient être prises en compte dans l'architecture proposée.

D'autre part, l'interopérabilité assurée dans SCEP-SOA a été illustrée par un exemple mettant en jeu deux applications d'entreprise différentes dédiées à la planification. Il serait intéressant d'élargir le périmètre d'application et d'implémentation de SCEP-SOA pour des cas faisant intervenir d'autres systèmes ayant des fonctionnalités autres que la planification et l'ordonnancement, *i.e.* applications CAD/CAM, PLM/PDM, CRM, etc.

* <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>

** <http://www.wsmo.org/2004/d2/v1.0/>

Références bibliographiques

A

(Abolhassani *et al.*, 2006) : H. Abolhassani, B. B. Hariri, and S. H. Haeri, "On ontology alignment experiments", *Webology*, Vol. 3, n° 3, article 28, September 2006. Available at: <http://www.webology.ir/2006/v3n3/a28.html>.

(Aleksovski *et al.*, 2006): Aleksovski Z., Klein M., Ten Kate W., Van Harmelen F, "Matching Unstructured Vocabularies using a Background Ontology", *Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW'06)*, Springer-Verlag, pp. 182-197, October 2006.

(Archimède et Coudert, 2001a) : B. Archimède and T. Coudert, "Ordonnancement multi-site des systèmes de production base sur le modèle SCEP", *APII-Journal Européen des Systèmes Automatisés*, 2001.

(Archimède et Coudert, 2001b): B. Archimède and T. Coudert, "Reactive Scheduling Using a Multi-Agent Model: The SCEP Framework", *Engineering Applications of Artificial Intelligence*, Vol. 14 n° 5, pp 667-683, 2001.

(Archimède *et al.*, 2003): Archimède, B., P. Charbonnaud, and C. Firmin, "A Supervised Multi-Site Reactive Production Activity Control Method for Extended Enterprise", in *Journal of Decision Systems*, Vol. 12 n° 3,4, pp. 309-328, 2003.

B

(Bal et Swift, 2002): Bal, J. and Swift, M., "Supporting SMEs through e-business", *manufacturing engineer*, pp. 219-224, 2002.

(Baumgärtel *et al.*, 1996): H. Baumgärtel, S. Bussmann, M. Klosterberg and D. Ag, "Combining Multi-Agent Systems and Constraint Techniques in Production Logistics", In *Proceedings of Artificial Intelligence, Simulation, and Planning in High Autonomous Systems*, pp. 361-367, 1996.

(Borst, 1997): W. N. Borst, "Construction of engineering ontologies for knowledge sharing and reuse", PhD thesis from University Twente, 5 september 1997.

(Botta-Genoulaz, 2005) : Botta-Genoulaz V., "Principes et Méthodes pour l'Intégration et l'Optimisation du pilotage des Systèmes de Production et des Chaînes Logistiques", Rapport de HDR, tome 1, Institut National des Sciences Appliquées de Lyon et Université Claude Bernard de Lyon 1, 2005.

(Botta-Genoulaz et Millet, 2005) : Botta-Genoulaz V. et P. A. Millet, "A classification for better use of ERP systems", *Computers in Industry*, Vol. 56, Issue 6, pp. 572-586, 2005. (Bouquet *et al.*, 2005): Bouquet, P., Euzenat, J., Franconi, E., Serafini, L., Stamou, G., and Tessaris, S., "Specification of a common framework for characterizing alignment", Deliverable 2.2.4, KnowledgeWeb, 2005.

(Bourrières *et al.*, 2005) : J.P. Bourrières, B. Grabot et C. Mercé, "Pilotage des systèmes multisites de production : outils industriels et méthodes avancées", *Techniques de l'Ingénieur*, S 7662, 15 pages, Septembre 2005.

(Breitman *et al.*, 2005): K. K. Breitman, A. H. Filho, E. H. Haeusler, and A. von Staa, "Using ontologies to formalize Services Specifications in Multi-Agent Systems", *Formal approaches to agent-based systems*, pp. 92-110, 2005, ISBN: 978-3-540-24422-6.

(Bussler, 2003): C. Bussler, "The Role of Semantic Web Technology in Enterprise Application Integration", in the Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 26 n° 4, pp. 62-68, 2003.

C

(Charlet *et al.*, 2000) : J. Charlet, M. Zacklad, G. Kassel et D. Bourigault, "Ingénierie des connaissances: Evolution récentes et nouveaux défis", Eyrolles, 2000.

(Chen *et al.*, 2006): D. Chen, M. Dassiti and B. Elvesaeter, "Enterprise Interoperability Framework and knowledge corpus", Research report of INTEROP NoE, FP6 Network of Excellence, Contract n° 508011, Deliverable DI.2, 2006.

(Corcho, 2002): O. Corcho, "A framework to solve the ontology translation problem", 2002.

(Curbera *et al.*, 2001): F. Curbera, W. A. Nagy, and S. Weerawarana, "Web services: Why and how?", In OOPSLA 2001 Workshop on Object-Oriented Web Services, ACM, 2001, available on line at: <http://www.research.ibm.com/people/b/bth/OOWS2001/nagy.pdf>.

D

(De Bruijn *et al.*, 2006): J. de Bruijn, M. Ehrig, C. Feier, F. Martin-Recuerda, F. Scharffe, and M. Weiten, "Ontology mediation, merging and aligning. In Semantic Web Technologies",

published by Wiley, 2006, available online at:
<http://www.debruijn.net/publications/mediation-chapter.pdf>.

(Deerwester *et al.*, 1990): Deerwester, S., Dumais, T. S., Furnas, W. G., Landauer, K. T., Harshman, R., "Indexing by Latent Semantic Analysis", In Journal of the American Society of Information Science, 1990.

(Dou *et al.*, 2002): Dou, D., D. McDermott, and P. Qi, "Ontology translation by ontology merging and automated reasoning", In Proc. EKAW2002 Workshop on Ontologies for Multi-Agent Systems, pp. 3–18, 2002.

(Dou *et al.*, 2005): D. Dou, D. McDermott, and P. Qi, "Ontology Translation on the Semantic Web", In Journal on Data Semantics II, LNCS 3360, pp. 35–57, 2005.

E

(Ehrig et Staab, 2004): M. Ehrig and S. Staab, "QOM – Quick Ontology Mapping", in S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 683–697, 2004.

(Enjalbert, 2006): S. Enjalbert, "Méthode et outil de simulation distribuée pour l’accommodation de la conduite de réseaux d’entreprises", Thèse de Doctorat soutenue publiquement le 19 Octobre 2006 à l’Institut National Polytechnique de Toulouse (INPT).

(Eurochambres, 2002): Eurochambres, association of European Chambers of Commerce and Industry, Draft Working Document of DG Enterprise B2B Internet platforms: Opportunities and barriers for SMEs- A first Assessment, August 2002.

(Eynard *et al.*, 2004): Eynard, B., Gallet, T., Nowak, P., and Roucoules, L., "UML based specifications of PDM product structure and workflow", *Computers in Industry*, Vol. 55, n°3, p. 301–316, 2004.

F

(Ferber, 1999): J. Ferber, "Multi-agent systems - an introduction to distributed artificial intelligence", Addison Wesley Longman, ISBN-10: 0201360489, 1999.

(FIPA, 2000): FIPA, 2000, FIPA Contract Net Interaction Protocol Specification, Foundation for Intelligent Physical Agents (FIPA).

(François, 2007): François, J., "Planification des chaînes logistiques: Modélisation du système décisionnel et performance", Thèse de Doctorat à l’Université Bordeaux I, 17 décembre 2007.

(Frankovic *et al.*, 2002): Frankovic, B., Budinska, I., Dang, T.T., "Creation of ontology for planning and modelling", Proceedings of the 3th International Symposium of Computational Intelligence, Budapest Hungary, pp. 73-84, 2002.

G

(Génin *et al.*, 2005): Génin, P., S. Lamouri, et A. Thomas, "Planification industrielle et ses limites", Techniques de l'ingénieur, 2005.

(Ghenniwa et Chen, 2004): Ghenniwa, H. and Chen W., "Service oriented e-marketplace: an agent-based model", Production planning and control, Vol. 15 n° 17, pp. 696-709, 2004.

(Gomez-Perez, 2000) : Gomez-Perez A., "Développements récents en matière de conception, de maintenance et d'utilisation d'ontologies", 3èmes rencontres Terminologie et intelligence artificielle TIA, Vol 19, pp. 9-20, 2000.

(Griffin et Pesch, 2007): Griffin, D. and Pesch, D., "A service oriented marketplace for next generation networks", Proceedings of the 6th international joint conference on Autonomous agents and multi-agent systems, Honolulu, Hawaii, ISBN: 978-81-904262-7-5, pp. 735-737, 2007.

(Gruber, 1993): T.R. Gruber, "A translation approach to portable ontologies", knowledge Acquisition, Vol. 5 n°2, p.p. 199-220, 1993.

(Gruber, 1995): Gruber T, "Toward principles for the design of ontologies used for knowledge sharing", 1995.

(Guarino 1997): Guarino N., "Understanding, building, and using ontologies", International journal of Human and Computer Studies, Vol. 45 n° 2,3, pp. 293-310, 1997.

(Guarino 1998): Guarino N., "Formal Ontology in Information Systems", Proceedings of FOIS'98, Trento, Italy, pp. 3-15, 1998.

(Guarino 1999) : Guarino N., "Some Ontological Principles for Designing Upper Level Lexical Resources", Proceedings of First International Conference on Language Resources and Evaluation, ELRA - European Language Resources Association, Granada, Spain, pp. 527-534, 1999.

(Gulledge, 2001): T. Gulledge, "B2B e-Marketplaces and small-and medium-sized enterprises", in proceedings of the Fourth International Conference on Stimulating Manufacturing Excellence in Small and Medium Enterprises, Aalborg, Denmark, 2001.

H

(Hasselbring, 2000): W. Hasselbring, "Information System Integration", Communications of the ACM, Vol. 43 n° 6, pp. 33-38, 2000.

(Hu, 2003): M. Hu, "Web services composition, partition, and quality of service in distributed system integration and re-engineering", XML Conference & Exposition, December 7-12, 2003, Pennsylvania Convention Center, Philadelphia, PA, USA, available on line at: http://www.idealliance.org/papers/dx_xml03/papers/05-05-04/05-05-04.html.

I

(Ishak *et al.*, 2008a): K. Ishak, B. Archimède, et P. Charbonnaud, "Modèle de planification distribuée interopérable de projets multi-sites basé sur une architecture orientée services", 9^{ième} Congrès EDSYS, 22 Mai 2008, à l'INSA de Toulouse France.

(Ishak *et al.*, 2008b): K. Ishak, B. Archimède, and P. Charbonnaud, "Applicative architecture for interoperable distributed planning of multi-site projects", proceedings of the 5th International Conference on Digital Enterprise Technology, 22-24 October 2008, Nantes France, ISBN 978-2-7483-4598-8, p.p. 193-209.

(Ishak *et al.*, 2008c): K. Ishak, B. Archimède, and P. Charbonnaud, "SCEP-SOA: An applicative architecture to enhance interoperability in multi-site planning", proceedings of the 3rd International Workshop on Enterprise Integration, Interoperability and Networking, November 12th-13th, 2008, Monterrey Mexico, LNCS 5333, p.p. 313-322, © Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-88874-1.

(Ishak *et al.*, 2008d): K. Ishak, B. Archimède, and P. Charbonnaud, "Enhancing interoperability between enterprise planning applications: an architectural framework", proceedings of the 3rd International Conference on Digital Information Management, sponsored by IEEE, November 13-16 2008, University of East London, London U.K., ISBN 978-1-4244-2917-2.

(Ishak *et al.*, 2009a): K. Ishak, B. Archimède, et P. Charbonnaud, "Une architecture orientée service pour assurer l'interopérabilité dans la planification multi-site", 3^{ième} Journées Doctorales du GDR MACS, 17-18 Mars 2009, Angers France.

(Ishak *et al.*, 2009b): K. Ishak, B. Archimède, and P. Charbonnaud, "Integration of semantic interoperability in a distributed architecture of multi-site planning", proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), June 3-5 2009 Moscow Russia, ISSN 1474-6670.

(Ishak *et al.*, 2009c): K. Ishak, B. Archimède, et P. Charbonnaud, "Interopérabilité sémantique dans une architecture orientée services de planification multi-site", 8^{ième} Congrès International de Génie Industriel (CIGI'09), 10-12 Juin 2009, Bagnères de Bigorre France, ISBN 978-2-7466-0894-8.

(Ishak *et al.*, 2009d): K. Ishak, B. Archimède, and P. Charbonnaud, "Managing multi-site production projects by a distributed and interoperable architecture", proceedings of the 5th International Virtual Conference on Innovative Production Machines and Systems (IPROMS 2009), 6-17July 2009.

(Ishak *et al.*, 2010): K. Ishak, B. Archimède, and P. Charbonnaud, "Integration of SOA and ontologies in a multi-agent model for interoperable and distributed scheduling", 8th International Conference of Modelling and Simulation (MOSIM'10), 10-12 May 2010, Hammamet Tunisia.

(Izza, 2006) : S. Izza, "Intégration des systèmes d'information industriels, une approche flexible basée sur les services sémantiques", thèse de Doctorat soutenue publiquement le 20 Novembre 2006 à l'Ecole Nationale Supérieure des Mines de Saint-Etienne.

J

(Jagdev et Browne, 1998) : Jagdev, H.S. and J. Browne, "The extended enterprise- a context for manufacturing", Production planning and control, Vol. 9 n°3, pp. 216-229, 1998.

(Jiao *et al.*, 2006): J. R. Jiao, X. You, and A. Kumar, "An Agent-Based Framework for Collaborative Negotiation in the Global Manufacturing Supply Chain Network", Robotics and Computer-Integrated Manufacturing, Vol. 22 n° 3, pp 239-255, 2006.

(Jinghua et Wenjian, 2005): L. Jinghua and L. Wenjian, "An agent-based system for multi-project planning and scheduling", in proceedings of the IEEE International conference on Mechatronics Automation, Niagara Falls, Canada, 2005.

K

(Kalfoglou et Hu, 2005): Kalfoglou Y., Hu B., "Crosi Mapping System (CMS) Result of the 2005 Ontology Alignment Contest", K-Cap'05 Integrating Ontologies workshop, Banff, Canada, pp. 77-85, 2005.

(Kart *et al.*, 2006): F. Kart, Z. Shen, C. E. Gerede, "The MIDAS system: A service oriented architecture for automated supply chain management", IEEE International Conference on Services Computing (SCC'06), p.p. 487-494, ISBN: 0-7695-2670-5, 2006.

(Kotis et Vouros, 2004): Kotis, K.; Vouros, G., "The HCONE Approach to Ontology Merging", In: *The Semantic Web: Research and Applications. First European Semantic Web Symposium ESWS 2004*, pp. 137-151.

(Kovalchuk, 2008): Y. Kovalchuk, "Multi-Agent Decision Support System for Supply Chain Management", *Proceedings of the 10th International Conference on Electronic Commerce (ICEC'08)*, Innsbruck, Austria, 2008.

(Kovacs et Paganelli, 2003): G. L. Kovacs and P. Paganelli, "A planning and management infrastructure for large, complex, distributed projects—beyond ERP and SCM", in *Computers in Industry*, Vol. 51, p.p. 165–183, 2003.

L

(Lazansky *et al.*, 2001): J. Lazansky, O. Stepankova, V. Marik, and M. Pechoucek, "Application of the multi-agent approach in production planning and modelling", *Engineering Application of Artificial Intelligence* Vol. 14, pp. 369-376, 2001.

(Li *et al.*, 2006): Li, Y., Lu, X., Chao, K. M., Huang, Y., and Younas, M., "The realization of service-oriented e-Marketplaces", *Information systems frontiers*, Vol. 8 n° 4, pp. 307-319, 2006.

(Lima *et al.*, 2006): R.M. Lima, R.M. Sousa and P.J. Martins, "Distributed Production Planning and Control Agent-Based System", *International Journal of Production Research*, Vol. 44, Nos. 18, 19, pp 3693-3709, 2006.

(LoNigro *et al.*, 2006): LoNigro, G. M. Bruccoleri and G. Perrone, "Negotiation in Distributed Production Planning Environments", *International Journal of Production Research*, Vol. 44 n° 18,19, pp. 3743-3758.

M

(Maedche, 2002): Maedche, A., "Ontology Learning for the Semantic Web", Kluwer Academic Publishers, 2002, in: K. K. Breitman, A. H. Filho, E. H. Haeusler, and A. von Staa, "Using ontologies to formalize Services Specifications in Multi-Agent Systems", *Formal approaches to agent-based systems*, pp. 92-110, 2005, ISBN: 978-3-540-24422-6.

(Malucelli, 2006): A. Malucelli, "Ontology-based services for agents interoperability", PhD Thesis at University of Porto, 2006.

(Marinova , 2002): Marinova Z. L. "Planning in multi-agent systems", MSc Thesis at Sofia University "St. Kliment Ohridski", 2002.

(Mizoguchi, 1995): Mizoguchi R., "Ikeda: Task Ontologies for reuse of Problem Solving Knowledge". In N. J. I. Mars (editors), *Towards Very Large Knowledge Bases*, IOS Press, 1995.

(Mohsenzadeh *et al.*, 2005): M. Mohsenzadeh, F. Shams, and M. Teshnehlab, "A New Approach for Merging Ontologies", In *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 4, February 2005 ISSN 1307-6884.

(Monsarrat *et al.*, 2005) : E. Monsarrat, C. Briand et P. Esquirol, "Aide à la décision pour une coopération inter-entreprise", RS- JESA- 39/2005, *Performance des chaînes logistiques*, p.p. 799-818.

(Mota *et al.*, 2002): L. Mota, J. Bento and L. Botelho, "Ontology definition languages for Multi-Agent Systems: the Geographical Information Ontology case study", *Proceedings of the Workshop on Ontologies in Multi-Agent Systems, 1st International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2002)*, Bologna Italy, 2002.

(Mota et Botelho, 2005): L. Mota and L. Botelho. "OWL Ontology Translation for the Semantic Web", 2005.

(Moutarlier *et al.*, 2000): Moutarlier, P., Geneste, L., Grabot, B., "Tapas: a modular framework to support reuse in scheduling software development", *Production Planning and Control*, Vol. 11 n° 7, pp. 648-659, 2000.

N

(Nagarajan *et al.*, 2006): M. Nagarajan, K. Verma, A. P. Sheth, J. Miller, and J. Lathem, "Semantic Interoperability of Web Services – Challenges and Experiences", *Proceedings of the IEEE International Conference on Web Services*, p.p. 373 – 382, ISBN:0-7695-2669-1, published by IEEE Computer Society Washington, DC, USA, 2006.

(Neches *et al.*, 1991): R. Neches, R. Fikes, T. Finin, T. Grubor, R. Patil, T. Senatir, and W. Swartout, "Enabling technology for knowledge sharing", *AI Magazine*, Vol. 12 n°3, p.p. 36-56, 1991.

(Nickull, 2005): D. Nickull, "Service Oriented Architecture (SOA) and Specialized Messaging Patterns", Adobe System Incorporated, technical white paper, 2005, <http://www.adobe.com/government/pdfs/SOA-technical-whitepaper.pdf>.

(Nishioka, 2004): Y. Nishioka, "Collaborative agents for production planning and scheduling CAPPS: A challenge to develop a new software system architecture for manufacturing

management in Japan”, International Journal of Production Research, Vol. 42 no 17, p.p. 3355-3368, 2004.

(Noy et Musen, 2000): Noy, N. F. and A. M. Musen, "PROMPT: Algorithm and tool for automated ontology merging and alignment", In Proceedings of the 17th National Conference On Artificial Intelligence (AAAI2000), Austin, Texas, USA.

(Noy et Musen, 2002): N. F. Noy and M. A. Musen, "Evaluating Ontology-Mapping Tools: Requirements and Experience", 2002.

(Noy, 2004): N. F. Noy, "Semantic integration: A survey of ontology-based approaches", SIGMOD Record, Vol. 33, n° 4, p.p. 65-70, December 2004.

O

(Obitko et Marik, 2002): M. Obitko and V. Marik, "Ontologies for Multi-Agent Systems in Manufacturing Domain", In Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02) 1529-4188/02, 2002.

(OECD, 2007): OECD Group, “Enhancing the role of SMEs in global value chains”, OECD background report, OECD global conference, 31 May-1 June 2007, Tokyo Japan.

(Oprea, 2005): Oprea, M., “A case study of agent-based virtual enterprise modelling”, Multi-Agent Systems and Applications IV, ISBN: 978-3-540-29046-9, pp. 632-635, 2005.

P

(Park et Yang, 2006): J. Park and J. Yang, ‘An International SME E-marketplace Networking Model’, in The Economics of Online Markets and ICT Networks, Physica-Verlag HD, ISBN978-3-7908-1706-5 (Print) 978-3-7908-1707-2 (Online), p.p. 245-257, 2006.

(Paviot, 2010): T. Paviot, “Méthodologie de résolution des problèmes d’interopérabilité dans le domaine du Product Lifecycle Management”, thèse de Doctorat à Supméca, Ecole Centrale de Paris, soutenue le 1 juillet 2010.

(Pires *et al.*, 2001): S. R. I. Pires, C. F. Bremer, L. A. De Santa Eualia and C. P. Goulart, “Supply Chain and Virtual Enterprises: Comparisons, Migration and a Case Study”, in International Journal of Logistics: Research and Applications, Vol. 4 n° 3, 2001.

(Pluempitiwiriyaew et Hammer, 2000): C. Pluempitiwiriyaew and J. Hammer, “A Classification Scheme for Semantic and Schematic Heterogeneities in XML Data Sources”, Technical Report TR00-004, University of Florida, Gainesville, FL, September 2000.

(Popovic, 2002): Popovic, M., "B2B e-marketplaces", Published by European Commission's Electronic Commerce Team (Information Society Directorate General), Brussels Belgium, June 2002.

R

(Ren *et al.*, 2006): Z. Ren, T. M. Hassan, C. D. Carter and C. J. Anumba, "E Contracting for SMEs through an engineering E-Hub", ITcon, Vol. 11, p.p. 161-173, 2006.

(Reynaud et Safar, 2006): Reynaud C., Safar B., "Structural Techniques for Alignment of Taxonomies: Experiments and Evaluation", In TR 1453, June 2006, LRI, Université Paris-Sud.

S

(Sabou *et al.*, 2006): Sabou M., D'Aquin M., Motta E., "Using the Semantic Web as Background Knowledge for Ontology Mapping", ISWC'06 Workshop on Ontology Matching (OM- 2006), Athens, Georgia, USA, 2006.

(Safar *et al.*, 2007): B. Safar, C. Reynaud, et F. Calvier, "Techniques d'alignement d'ontologies basées sur la structure d'une ressource complémentaire", 1^{ères} journées Francophones sur les Ontologies (JFO 2007), Sousse, Tunisie, 18-20 Octobre 2007.

(Saleem, 2006): A. Saleem, "Semantic Web vision: survey of ontology mapping systems and evaluation of progress", Master Thesis Intelligent Software Systems, School of Engineering Blekinge Institute of Technology, November 2006.

(Sauer *et al.*, 2000): J. Sauer, T. Freese and T. Teschke, "Towards Agent-Based Multi-Site Scheduling", Proceedings of the ECAI 2000 Workshop on New Results in Planning, Scheduling, and Design, 123-130, Berlin, 2000.

(Sauer et Appelrath, 2002): J. Sauer and H.-J. Appelrath, "Scheduling the Supply Chain by Teams of Agents", in Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03) 0-7695-1874-5/03, 2002.

(Singh et Huhns, 2005): Singh M., and Huhns M., "Service-Oriented Computing, Semantics, processes, agents", Wiley, ISBN: 978-0-470-09148-7, 2005.

(Smith et Becker, 1997): Smith, S. F. and M. A. Becker, "An ontology for constructing scheduling systems", Working Notes of AAAI Press, AAAI Symposium on Ontological Engineering, Palo Alto, CA, 1997.

(Smith, 1980): R. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver", IEEE Trans. on Comp Vol. 29, p.p. 1104-1113, 1980.

(SOA Glossary): <http://www.soaglossary.com>.

(Srivastava, 2003) : B. Srivastava, "Web service composition - current solutions and open problems", ICAPS 2003 Workshop on Planning for Web Services, pp. 28-35, 2003, available on line at: <http://www.informatik.uni-freiburg.de/~koehler/bpia/icaps-ws.pdf>.

(Studer *et al.*, 1998): R. Studer , V. R. Benjamins, and D. Fenrel, "Knowledge engineering principles and methods", IEEE Transaction on Data and Knowledge Engineering. Vol. 25 n°1,2, p.p. 161-197, 1998.

(Stumme et Maedche, 2001): G. Stumme and A. Maedche, "FCA-Merge: Bottom-Up merging of ontologies", In Proc. 17th Intl. Conf. on Artificial Intelligence (IJCAI '01), Seattle, WA, USA, pp. 225-230. 2001.

T

(Tang *et al.*, 2005): J. Tang, B.Y. Liang, and J. Z. Li. "Toward detecting mapping strategies for ontology interoperability", WWW 2005, May 10--14, 2005, Chiba, Japan.

(Telle, 2003) : Telle O., "Gestion des chaînes logistiques dans le domaine aéronautique : aide à la coopération au sein d'une relation donneur d'ordres – fournisseurs", Thèse de doctorat, SUPAERO de Toulouse, 2003.

(Thibaud *et al.*, 2007): M. Thibaud, R. Daniel, and A. Didier, "Multi-site coordination using a multi-agent system", Computers in Industry, Vol. 58 n° 4, pp. 367-377, 2007.

(Touaf, 2005): Touaf, S., "Diagnostic logique des systèmes complexes dynamiques dans un contexte multi-agent", Thèse de Doctorat à l'Université de Joseph Fourier-Grenoble 1, 02 mars 2005.

(Touzi, 2007) : J. Touzi, "Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises", thèse de Doctorat soutenue publiquement le 09/11/2007 à l'Institut National Polytechnique de Toulouse.

U

(Udomleartprasert *et al.*, 2003): P. Udomleartprasert, C. Jungthirapanich and C. Sommechai, "Supply Chain Management – SME Approach", 2003.

(Uschold et Gruninger 1996): Uschold M., and Gruninger M., "Ontologies: Principles, Methods and Applications". Knowledge Engineering Review, Vol. 11 n° 2, 1996.

V

(Van Heijst *et al.* 1997): Van Heijst G., Schreiber A. Th., and Wielinga B. J., "Using explicit ontologies in KBS development", International Journal of Human- Computer Studies, Vol. 46 n° 2,3 pp.183-292, 1997.

(Vernadat, 1996): F. B. Vernadat, "Enterprise modelling and integration: Principles and applications", Chapman & Hall, London, 1996.

(Vernadat, 2006): F. Vernadat, "Interoperable enterprise systems: architecture and methods", plenary lecture, IFAC/INCOM conference, Saint-Etienne, May 2006.

W

(Wache *et al.*, 2001) : Wache, H. , Vogele, T. , Visser, U. , Stuckenschmidt, H., Schuster, G., Neumann, H., and Hubner, S., "Ontology-based integration of information - a survey of existing approaches", IJCAI-01 Workshop on Ontologies and Information Sharing, pp. 108-117, 2001.

(Wong *et al.*, 2006): T.N. Wong, C.W. Leung, K.L. Mak and R.Y.K. Fung, "An agent based negotiation approach to integrate process planning and scheduling", International journal of production research (IJPR), Vol. 44 n° 7, pp. 1331-1351, 2006.

(Wongthongtham et Chang, 2004): P. Wongthongtham, E. Chang, "Ontology-based Multi-agent System to Multi-site Software Development", in proceedings of the Workshop QUTE-SWAP@ACM/SIGSOFT-FSE12, November 5, 2004, Newport Beach, CA, USA.

Z

(Zhou et Kuntz, 2004): J. Zhou and R. Dieng-Kuntz, "Manufacturing Ontology Analysis and Design: towards Excellent Manufacturing", supported by the ERCIM the European Research Consortium for Informatics and Mathematics fellowship, 0-7803-8513-6/04, 2004.